HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

BACHELORARBEIT

Herr
René Kaul

Konzeption und Realisierung einer offlinefähigen Webanwendung für Terminals und deren Aktualisierung bei vorhandener Internetanbindung

Mittweida, 2011

Fakultät Elektro- und Informationstechnik

BACHELORARBEIT

Konzeption und Realisierung einer offlinefähigen Webanwendung für Terminals und deren Aktualisierung bei vorhandener Internetanbindung

Autor:
Herr René Kaul

Studiengang:
Multimediatechnik

Seminargruppe:
MK07w1

Erstprüfer:
Prof. Dr.-Ing. Frank Zimmer

Zweitprüfer:
Dipl.-Inf. (BA), Stephan Preuß

Einreichung:
Mittweida, 01.11.2011

Verteidigung/Bewertung:
Mittweida, 2011

Bibliografische Angaben:

René Kaul:
Konzeption und Realisierung einer offlinefähigen Webanwendung für Terminals und deren Aktualisierung bei vorhandener Internetanbindung - 2011 – 61 S.
Leipzig, Hochschule Mittweida (FH), University of Applied Sciences, Fakultät Elektro- und Informationstechnik, Bachelorarbeit, 2011

Referat:

Diese Bachelorarbeit befasst sich mit der Konzeption und Realsierung einer offlinefähigen Webanwendung. In der Konzeption werden notwendige Richtlinien und Vorgaben geklärt, sowie eine Prüfung der notwendigen Technologien ausgeführt. In Rahmen der Umsetzung wurde näher auf die Gestaltung, Implementierung und die Maßnahmen zur Offlinefähigkeit der Webanwendung eingegangen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

In den ersten Kapitelabschnitten der Bachelorarbeit, wird auf die Motivation und die Zielsetzung eingegangen. Zusätzlich wird ein Überblick über die Vorgehensweise in der Arbeit aufgezeigt.

1.1 Motivation

Motivation für die vorliegende Arbeit war es, eine verkaufsförderliche Webanwendung zu erstellen, die alle Altersklassen anspricht und gut bedienbar ist. Durch die Erzeugung einer auch offline lauffähigen Webanwendung, wird eine sichere Möglichkeit geboten, um Komplikationen wie z.B. durch ungewollte Internetabbrüche, wodurch die Webanwendung ausfällt, unterbunden werden.

1.2 Ziel der Arbeit

Ziel der Arbeit war die Konzeption und Realisierung einer Webanwendung. Die Webanwendung soll künftig auf Messen über Infoterminals genutzt werden. Bei der Konzeption war zu beachten, dass die Corporate Identity der Firma übernommen wurde und eine benutzerfreundliche Darstellung für die Anwender gewährleistet ist. Als Betriebsmedium wird künftig ein Terminal mit Touch-Bedienung verwendet.

Hinzu soll eine geeignete Lösung gefunden werden, die es der Webanwendung ermöglicht, bei nicht vorhandener Internetanbindung, trotzdem Inhalte darzustellen. Bei vorhandener Internetanbindung soll eine Aktualisierung von veränderten Daten erfolgen. Primär sollen Inhalte, wie Bilder, Texte und veränderte HTML-Inhalte aktualisiert werden. Dafür musste eine geeignete Technologie gefunden werden, mit denen die Webanwendung realisiert werden kann.

1.3 Vorgehensweise in der Arbeit

Phase 1: Organisation und Konzeption

In dieser Phase wird mit den Kunden interagiert, um notwendige Details für die Entwicklung festzulegen. Darunter zählt die Abklärung über notwendige Inhalte und Grafiken, Gestaltungs- bzw. Darstellungsrichtlinien und das festlegen von Funktionalitäten der Webanwendung. Mehrere Kundenmeetings während der Konzeption sollen die Zusammenarbeit von Kunden und Produzenten erleichtert. Um kommende Unklarheiten vorzeitig aus dem Weg zu räumen. Am Schluss muss eine Übersicht über notwendige Richtlinien und Vorgaben vorhanden sein. Auch eine kurze Erklärung notwendiger Grundlagen soll erfolgen.

Phase 2: Umsetzung der Webanwendung

In Phase 2 kommen gestalterische Fähigkeiten zum Einsatz. Hierbei muss beachtet werden, dass die Corporate Identity bei der Gestaltung mit einfließt. Vorab sollen die sogenanntem Scribble Skizzen entworfen werden und ggf. durch Mockups ergänzt werden. Fertige Entwürfe werden mit dem Arbeitgeber durchgegangen, und sobald ein gemeinsamer Nenner gefunden wurde, kommt es zur digitalen Umsetzung. Danach werden Layouts auf Basis, der vorher erstellten Entwürfe angefertigt, welche farblich mit der CI übereinstimmen müssen. Nachdem das Grundlayout vorhanden ist, werden die einzelnen Unterseiten gestaltet. Durch die Absegnung durch den Auftraggeber wird die Implementierung der Anwendung eingeleitet.

Als erstes muss das ausgewählte Content Management-System, in einer Testumgebung eingerichtet werden. Danach werden notwendige, Templates und Module für das Frontend, dem Content Management-System hinzugefügt. Danach sollen die HTML Elemente mittels CSS, formatiert werden. Nachdem die Webanwendung erstellt wurde, sollen Versuche mit der Offlinefähigkeit durchgeführt werden.

Phase 3: Anwendungstest

Zum Abschluss wird die Anwendung auf Herz und Nieren getestet. Dabei wird überprüft ob alle notwendigen Richtlinien und Vorgaben, ordentlich umgesetzt wurden. Falls notwendige Änderungen bestehen, werden diese noch beseitigt.

2 Konzeption Webanwendung

In diesem Kapiteln werden notwendige Richtlinien und Vorgaben für die Umsetzung festgelegt, notwendige Technologien werden geprüft und vorgestellt.

2.1 Richtlinien und Vorgaben

Zur Orientierung für die Umsetzung wurden folgende Richtlinien und Vorgaben festgelegt.

Da Inhalte dynamisch über die Redakteure erzeugt oder bearbeitet werden können. Muss als Grundlage der Webanwendung ein Content Management-System verwendet werden. Eine vorige Prüfung geeigneter Content Management-Systeme soll die Wahl erleichtern. Nach der Prüfung muss das gewählte CMS eingerichtet werden. Alle Module und Templates, die dem CMS hinzugefügt werden, sollen namentlich eindeutig gekennzeichnet werden. Die Ablegung von Dateien soll separat zum CMS erfolgen und in einer übersichtlichen Ordnerstruktur. Soweit Erweiterungen für das Content Management-System verwendet werden, sollen diese ordnungsgemäß eingebaut werden. Wenn Inhalte in einer Datenbank abgespeichert werden müssen, soll hierfür eine MySQL-Datenbank verwendet werden.

Der Umfang der Webanwendung im Frontend soll vier Inhaltsseiten betragen, die jeweils ein anderes Themengebiet darstellen. Folgende Inhaltsseiten sollen vorhanden sein, zwei Referenzenseiten, die Videos oder Bilder darstellen. Eine Informationsseite, die Informationen über Beiträge des HANDSPIEL-Usability-Tag 2011 bietet, diese Seite soll gleichzeitig auch als Startseite verwendet werden. Die letzte und vierte Seite wird dazu verwendet, aktuelle Sponsoren des HANDSPIEL-Usability-Tags 2011 vorzustellen.

Folgende Funktionalitäten sollen auf den einzelnen Inhaltsseiten verfügbar sein. Da auf der Beitragsseite nicht alle Beiträge, aus Platzgründen

gelistet werden können, muss ein Wechsel zwischen dem vorigen und nächsten Beitragen ermöglicht werden. Bei der Darstellung von Videos und Bildern muss ein Wechsel zwischen den einzelnen Medien möglich sein. Für die Videoseite muss ein geeigneter Player gefunden werden, über den man das Video starten und pausieren kann. Funktionen wie eine Lautstärkeregelung und eine Fortschrittsanzeige wären zusätzlich von Vorteil.

Allgemeine Funktionen, die für jede Inhaltsseite der Webanwendung verfügbar sein müssen, wären. Es muss eine Navigation erstellt werden die ein Wechsel, zwischen den einzelnen Inhaltsseiten ermöglicht. Eine eindeutige Identifizierung der unterschiedlichen Informationsseiten, kurz gesagt die aktuelle Kategorie muss erkennbar sein. Und damit keine aktuellen Vorträge verpasst werden, muss die aktuelle Zeit im Frontend sichtbar sein. Eine weitere Funktion ist, dass die Webanwendung bei nicht vorhandener Internetanbindung, trotzdem funktionsfähig bleibt. Hierfür müssen Inhaltsdaten, wie z.B. Bildern und Texte auf den ausführenden Computer gespeichert werden. Sobald eine Aktualisierung über das Internet erfolgt, müssen die bisherigen Daten überschrieben werden.

Bei der Darstellung der Webanwendung wird eine FullHD Touchscreen verwendet. Somit darf die Webanwendung eine Seitengröße von 1920x1080 Pixel nicht überschreiten, um störende Scrollbalken zu vermeiden. Bei der Wiedergabe eines Videos soll ein Verhältnis von 4:3 eingehalten werden. Eine weitere Festlegung ist, dass die Darstellung der Webanwendung, nur über einen Firefox Browser erfolgen soll. Anpassungen für andere Browser entfallen somit. Als Betriebssystem wurde Windows 7 festgelegt. Des Weiteren soll die Webanwendung auf einen Webserver abgelegt werden, der Technologien wie PHP und MySQL unterstützt.

Damit eine optische Identifikation der Webanwendung mit der Handspiel GmbH, durch den Nutzer ersichtlich ist, müssen verwendete Grafiken und Farben, die zum Beispiel auf der Handspiel Webseite verwendet wurden, bei der Gestaltung einfließen.

Bei der Entwicklung der Webanwendung soll darauf geachtet werden, dass aktuelle und bekannte Technologien verwendet werden, wie z.B. PHP, HTML, JavaScript und jQuery. Informationen zu den Technologien, sind im Kapitel 2.4 zu finden.

2.2 Inhaltsanalyse

Damit bei der grafischen Umsetzung, alle notwendigen Inhalte bekannt sind, wird in diesem Kapitel, gezielt auf feste und mögliche Inhalte eingegangen.

Beitragsseite

Um einen Überblick zu erhalten, welche Beiträge am Usability-Tag vorhanden sind, wird eine Übersichtsseite für die festgelegten Beiträge benötigt. Ein Beitragseintrag muss über das aktuelle Thema, den Referenten, Ort, Zeit und den Veranstalter informieren. Da nicht alle Einträge gleichzeitig darstellbar sind, wird die Anzahl der sichtbaren Einträge, drei bis maximal fünf sein. Sobald diese überschritten ist, muss ein Wechsel zwischen dem vorigen und nächsten Beitrag, ermöglicht werden.

Beispielseite

Eine weitere Seite wird benötigt, um bereits erstellte Produkte, von der Handspiel GmbH zu präsentieren. Inhaltlich soll die Seite, wie eine Bildergalerie aufgebaut sein. Das heißt ein Detailbild und eine Navigation zwischen den einzelnen Beispielen. Um weitere Informationen über das aktuelle Bild zu erhalten, wird zusätzlich der zugehörige Titel und Beschreibung, mit dargestellt.

Videoseite

Damit Videos wiedergegeben werden können, wird ein Videoplayer benötigt. Der über eine Navigationsleiste verfügt, mit folgenden Funktionen, starten, pausieren, Lautstärkeregelung und Fortschrittsanzeige. Unterhalb des Players befindet sich eine Navigation, in der möglich ist, andere Videos auszuwählen. Mittels Vorschaubild und

Titel soll erkennbar sein, über welche Inhalte die Videos verfügen. Insgesamt sollen nur drei Videos zur Auswahl stehen.

Sponsorenseite

Die letzte Seite wird dazu verwendet, um alle Sponsoren des HANDSPIEL Usability-Tages aufzulisten. Folgende Daten sollen präsentiert werden, Firmenname, Adresse und Logo.

2.3 Auswahl Content Management-System

Damit eine problemlose Umsetzung der Webanwendung möglich ist, wurden folgende Anforderungen an das Content Management-System gestellt:

- Komplikationsfreie und umstandsfreie Realisierung einer Webseite
- Einfache Inhaltsverwaltung (Inhalte bearbeiten, löschen, hinzufügen)
- Benutzerverwaltung
- Modularer Aufbau zur einfachen Erweiterung
- Geringer Schulungsaufwand
- Aktive Community
- Open Source

Aus betrieblichen und redaktionellen Erfahrungen sind folgende zwei CMS in die engere Auswahl gekommen:

REDAXO

„REDAXO vereint hohe Flexibilität mit einfacher Handhabung für sinnvolle Nutzung. Es eignet sich sowohl für kleinere Auftritte als auch für große und komplexe Internetportale. Dank des modularen Aufbaus und der vielen Erweiterungsmöglichkeiten deckt REDAXO alle erforderlichen Funktionalitäten eines umfassenden Redaktionssystems ab. Zusätzlich ist

REDAXO ein Open-Source-System und somit kostenlos und kommerziell frei verwendbar.“ (5)

TYPO3

TYPO3 ist ein frei konfigurierbares Content Management-System zur Pflege von dynamisch generierten Internetpräsentationen, das in den letzten Jahren zunehmend an Bedeutung gewonnen hat. TYPO 3 ist Open Source und wurde unter der GPL-Lizenz[1] veröffentlicht. TYPO3 beherrscht neben der Ausgabe im HTML-Format auch Exporte in andere Formate, beispielsweise in XML, PDF usw. Die Idee für die Entwicklung stammt von Kasper Skaarhoj. (7)

Entscheidung

Beide CMS verfügen über eine gute Community und somit über ausreichend Hilfe bei Problemen. Auch vorgefertigte Module, Add-ons oder Extensions, die von Nutzern erstellt wurden, stehen zur Verfügung. Die Dokumentation ist bei beiden CMS verfügbar. Gegenüber REDAXO, fällt die Dokumentation bei Typo3 umfangreicher aus, ist aber sehr technisch und meistens in Englisch verfasst. Damit unterschiedliche Nutzer auf das Backend zugreifen können, bieten auch beide CMS, steuerbare Zugriffsrechte an.

Letztendlich wurde sich, für das REDAXO-CMS entschieden, da die benötigten Anforderungen erfüllt sind und bereits Erfahrungen, mit diesen CMS gesammelt wurden. Folgende Dinge sprachen gegen TYPO3, die Erlernung der Scriptsprache TypoScript und die nicht Notwendigkeit eines so umfangreichen CMS.

[1] General Public License – Freie Software Lizenz

2.4 Webtechnologien

2.4.1 HTML & CSS

HTML – HyperText Markup Language

„HTML ist eine sogenannte Auszeichnungssprache. Sie hat die Aufgabe, die logischen Bestandteile eines textorientierten Dokuments zu beschreiben. Als Auszeichnungssprache bietet HTML daher die Möglichkeit an, typische Elemente eines textorientierten Dokumentes, wie Überschriften, Textabsätze, Listen. Tabellen oder Grafikreferenzen, als solche auszuzeichnen.

Das Auszeichnungsschema von HTML geht von einer hierarischen Gliederung aus. HTML zeichnet Inhalte von Dokumenten aus. Dokumente haben globale Eigenschaften wie zum Beispiel Kopfdaten. Der eigentliche Inhalt besteht aus Elementen, zum Beispiel einer Überschrift 1. Ordnung, Textabsätzen, Tabellen und Grafiken. Einige dieser Elemente haben wiederum Unterelemente. So enthält ein Textabsatz zum Beispiel eine als betont oder fett markierte Textstelle, eine Aufzählungsliste besteht aus einzelnen Listenpunkten, und eine Tabelle gliedert sich in einzelne Tabellenzellen.

Die meisten dieser Elemente haben einen fest definierbaren Erstreckungsraum. So geht eine Überschrift vom ersten bis zum letzten Zeichen, eine Aufzählungsseite vom ersten bis zum letzten Listenpunkt oder eine Tabelle von der ersten bis zur letzten Zelle. Auszeichnungen markieren Anfang und Ende von Elementen. Aktuellste Version von HTML 4.01, veröffentlicht durch das World Wide Web Consortium, welches den Webstandard festgelegt hat.“ (8)

HTML5

„Durch die Browser-Hersteller Opera Software ASA, die Mozilla Foundation und die Apple Inc. wurde eine Arbeitsgruppe mit dem Namen "Web Hypertext Application Technology Working Group" (WHATWG) gegründet, die es zur Aufgabe hat, neue, moderne Webtechnologien zu entwickeln. Diese Technologien werden unter dem Begriff "HTML5"

entwickelt. Das W3C, unter Leitung von Berners-Lee, hatte parallel das Ziel, den XHTML2-Standard weiter zu entwickeln, dies wurde jedoch im Jahr 2009, aufgrund negativer Zukunftsaussichten des Standards, eingestellt. Aktuell entwickeln beide Arbeitsgruppen gemeinsam am HTML5-Standard. Mit HTML 5 wurden die bestehenden Versionen und Standards HTML 4.01, XHTML sowie das Document Object Model ersetzt. Aktuell ist HTML in der Version 5 noch nicht endgültig definiert und hat somit lediglich „Entwurfs“-Charakter. HTML 5 wird seit 2007 entwickelt und stellt den aktuellen Stand der HTML-Entwicklung dar. Eine besondere Neuerung hierbei ist, dass mit dieser Version Abstand von der SGML-Typisierung der HTML-Sprache genommen wurde und stattdessen die Sprache nun als Document Object Model (DOM) beschrieben ist. Besondere Neuerungen an HTML 5 sind die implementierten zusätzlichen Möglichkeiten, mit multimedialen Elementen zu arbeiten. Es werden nun Audio und Video und dynamische Grafiken direkt unterstützt, bei denen in älteren HTML Versionen das Zurückgreifen auf andere Techniken, abseits von HTML, notwendig wurde.“ (4)

CSS - Cascading Stylesheets

„Da HTML den Grundsätzlichen Aufbau ihrer Webseite bestimmt und nicht für das Aussehen der Elemente zuständig ist. An diesem Punkt setzen die Cascading Stylesheets (CSS) ein. Es handelt sich dabei um eine unmittelbare Ergänzungssprache, die vorwiegende für HTML entwickelt wurde. Sie klinkt sich nahtlos in HTML ein und erlaubt das beliebige Formatieren einzelner HTML-Elemente. Mit Hilfe von CSS Stylesheets können Sie beispielweise festlegen, das alle Überschriften 1. Ordnung 24 Punkt groß sind, in roter Helvetika-Schrift, mit einem Nachabstand von 16 Punkt und mit einer grünendoppelten Rahmenlinie oberhalb dargestellt werden. Sie können genauso gut auch für einen beliebigen Text festlegen, dass nur dieser Text 3 Zentimeter groß sein soll und eine gelbe Hintergrundfarbe erhält. Daneben enthält CSS auch die Möglichkeiten zum punktgenauen Platzieren von Elementen am Bildschirm und für andere Ausgabemedien wie Drucker oder Audio-System.

CSS ist ebenso wie HTML eine Klartextsprache. Auch für CSS brauchen Sie keine bestimmte Software, es genügt ein Texteditor. CSS ist wie HTML eine offen dokumentierte und vom W3 Konsortium standardisierte Sprache, dies Sie frei und ohne Lizenzprobleme verwenden können.“(8)

In CSS wird zwischen drei Versionen unterschieden, erkenntlich durch die aufsteigende Nummerierung am Ende, wobei die Versionsnummer ausgeschrieben z.B. Level 1 heißt.

CSS1 und CSS2, sind beide anerkannte Webstandards und werden somit in allen gängigen Browsern unterstützt. Zu der Palette der Eigenschaften, gehören z.B. das Positionieren von Elementen sowie Schriftanpassungen.

CSS3 gehört noch nicht zu den Webstandards, trotzdem unterstützen viele Browser diese Technologie. Auch für Browser die CSS3 nicht unterstützen, gibt es geeignete Workarounds[1] . Durch CSS3 können jetzt zum Beispiel Verläufe oder Schatten zu Elementen hinzugefügt werden, die früher nur mit Grafiken realisiert werden konnten.

2.4.2 PHP & MySQL

PHP - Hypertext Preprocessor

PHP ist die Abkürzung für "*PHP: Hypertext Preprocessor*", eine weitverbreitete Open Source Skriptsprache speziell für Webentwicklungen. PHP lässt sich in HTML einbinden. Die Syntax erinnert an C, Java und Perl und ist einfach zu erlernen. Das Hauptziel dieser Sprache ist es, Webentwicklern die Möglichkeit zu geben, schnell dynamisch generierte Webseiten zu erzeugen. Aber Sie können PHP für weitaus mehr einsetzen. (9)

Die Integration von PHP, kann über mehrere Schreibweisen eingebunden werden. Üblicherweise wird die XML[2]-Schreibweise verwendet.

```
<?php echo „Einbindung in XML-Stil“; ?>
```

[1] Workarounds - Notlösungen
[2] XML – Extensible Markup Language

MySQL

Ist ein Datenbankmanagement System. Es bietet die Grundlage für viele dynamische Webauftritte. Es werden Datenbanken zu Verfügung gestellt, in denen Inhalte mittels Tabellen abgespeichert werden. Um auf die Daten zu zugreifen, kann zum Beispiel die Scriptsprache PHP verwendet werden.

2.4.3 JavaScript & jQuery

JavaScript

„JavaScript ist eine zur Erweiterung des HTML-Befehlssatzes entwickelte kompakte Scriptsprache, die es auch Entwicklern mit geringen Programmierkenntnissen ermöglicht, objektorientierte Anwendungen in Internetseiten zu implementieren. Sie wird direkt in das HTML-Dokument eingebunden, wodurch schnelle Aktionszeiten möglich sind. Um JavaScripte entwickeln zu können, sind keine zusätzlichen Editoren wie AppBuilder, Compiler, Debugger oder sonstige Entwicklungsumgebungen nötig. Es kann wie HTML-Code einfach in Texteditoren eingegeben werden.“(3)

jQuery

Entwickelt wurde jQuery von John Riesig, dabei handelt es sich um ein frei verfügbares JavaScript-Framework. Folgende Funktionen werden von jQuery zur Verfügung gestellt:

DOM Manipulation, können Inhalte (zum Beispiel Elemente) geändert, gelöscht oder hinzugefügt werden. JQuery hat dafür, mehrere Methoden zur Verfügung, wie z.B. `.after(inhalt)`, diese Methode fügt einen bestimmten Inhalt, nach dem selektierten Element ein.

Für Animationen stellt jQuery grundlegende Methoden(Effects) zur Verfügung, wie zum Beispiel Ein- und Ausblenden. Zeigen und Verstecken oder Fades.

Um gezielt Elemente oder Elementgruppen anzusprechen, werden in jQuery Selektoren genutzt, diese ähneln den Selektoren, die auch in CSS, verwendet werden. Selektoren werden in folgende Gruppen aufgeteilt:

- Basisselektoren
- Mehrfachklassenselektor
- Gruppen- und Kontextselektoren

Zusätzlich können zu den Selektoren Filterausdrücke hinzugefügt werden. Weitere Filter Möglichkeiten wären, Inhaltsfilter, Sichtbarkeitsfilter, Attributfilter und Child-Filter.

Weitere Funktionen:

- Event Handling
- Form Handling
- Ajax und JSON
- Traversing
- Collect und Select

Damit die JavaScript-Bibliothek-jQuery verwendbar wird, muss die jQuery.js, auf der gewünschten Seite, wo jQuery zum Einsatz kommt, eingebunden werden. Das Integrieren sollte möglichst im Header geschehen. (10)

```
<script

     src="jQuery.js"

     type="text/javascript">

</script>
```

2.4.4 Prüfung verwendbarer Technologien auf Offlinefähigkeit

Damit die eine offlinefähige Anwendung, entstehen kann, musste in vorab geprüft werden, welche möglichen Technologien infrage kommen. Durch Recherchen in Büchern und im Internet wurden folgende zwei Technologien gefunden.

Gears

Gears ist ein Plug-In, wodurch der Browser erweitert wird und dadurch eine umfassende Plattform für Webanwendungen bietet. Webmaster können Gears beispielsweise auf ihren Websites nutzen, um Nutzern die Möglichkeit zu geben, offline auf Informationen zuzugreifen, oder um Content auf Basis Ihres geografischen Standorts bereitzustellen. (2)

Mittlerweile ist Stand der Dinge, das ab Dezember 2011, Gears nicht mehr zur Verfügung steht. Stattdessen wird Google Inc. sein Hauptaugenmerk auf HTML5 legen, um dies mit zu Verbessern und Standards festzulegen.

HTML5

Eine andere Möglichkeit um eine Webanwendung offlinefähig zu machen, ist mit der Webtechnologie HTML5.

So bietet HTML5 folgende Offline-Features:

- Offline Application Caching API (Application Cache)
- DOM[1] Storage und Online-/Offline Events
- Web SQL-Database: eine clientseitige JavaScript Datenbank

Da Gears ab Dezember 2011 nicht mehr zur Verfügung steht, wird bei der Umsetzung, HTML5 dazu verwendet, um einen Offlinefähigkeit der Webanwendung zu ermöglichen. In dem Kapitel 3.3 Maßnahmen zur Offlinefähigkeit, ist dann die Umsetzung zu finden.

[1] DOM - Document Object Model

3 Umsetzung der Webanwendung

3.1 Gestaltungskonzepte und digitale Umsetzung

Die Vorgehensweise bei der Gestaltung ist in Kapitel 2.1 festgelegt und die Inhalte für den Inhaltsbereich sind in Kapitel 2.2 zu finden.

3.1.1 Scribble Skizzen

In der kommenden Abbildung 3.1-1, werden ein paar ausgewählte Scribble Skizzen dargestellt. Scribble Skizzen dienen dazu um erste Vorentwürfe, für das kommende Layout zu erstellen. Dabei müssen die Inhalte nicht unbedingt sauber erstellt sein.

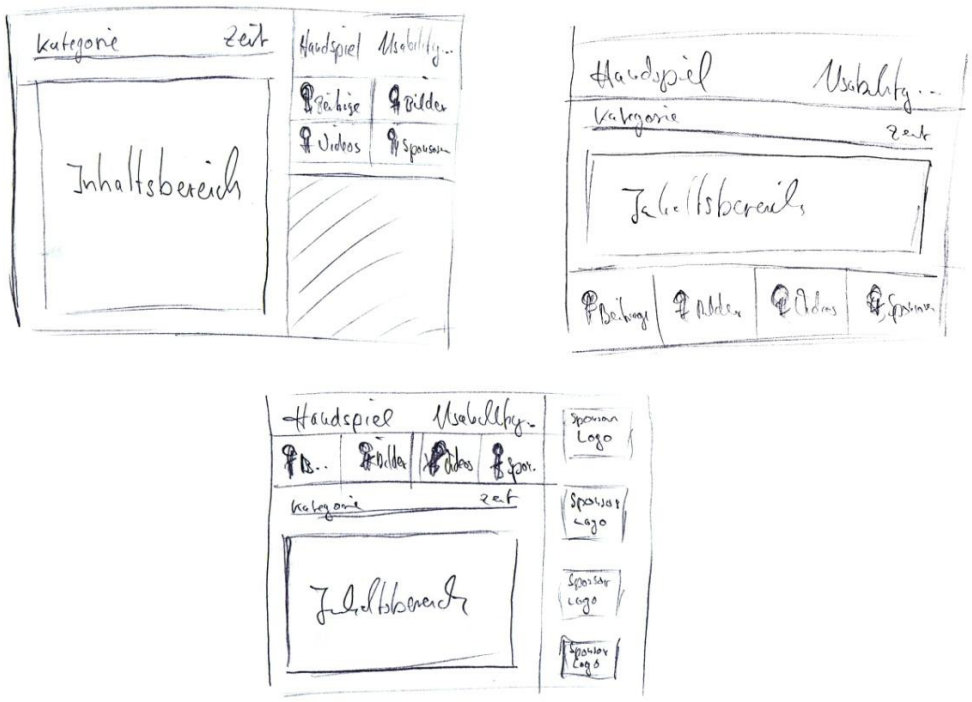


Abb. 3.1-1 Scribble Skizzen

Für das kommende Layout orientierte man sich an der Scribble Skizze die sich im linken oberen Bereich der Abb. 3.1-1, befindet. Diese Skizze bietet eine optimale Trennung von Navigation und Informationstexten zwischen den zu präsentierten Inhalten im linken Bereich.

3.1.2 Digitale Umsetzung

Wie festgelegt soll zur Orientierung, für die farblichen Richtlinien, die Handspiel-Webseite verwendet werden. Folgende Inhalte wurden identisch von der Handspiel-Webseite übernommen: die Hintergrundfarbe mit dem dazugehörigen transparenten Muster. Genauso die Grafik, die sich im Hintergrund am Fußende der Webseite befindet. Mehr dazu im Kapitel 3.1.2.4. Inhaltsbereiche werden mit einem weißen Hintergrund und zur Verbesserung der Tiefenwirkung, mit einem Schlagschatten versehen. Die Navigation soll farblich ungefähr, den Farbtönen der Handspiel Navigation entsprechen. Zusätzlich wird der Navigation, unterschiedliche Handspielmännel, für die jeweiligen Themenbereiche zugeordnet. Dadurch wird eine bessere Identifizierung mit der Handspiel GmbH gewährleistet.

3.1.2.1 Grundlayout

Für das Grundlayout wurde der Goldene Schnitt als Vorlage verwendet. Im unteren Beispiel ist jeweils ein 3:5 Verhältnis zu sehen, exakter formuliert, verhält sich die Gesamtstrecke zu der größeren, wie die größere zu der kleineren. Dies soll, laut Euklid, sehr harmonisch auf Menschen wirken.

Abb. 3.1-2 Aufteilung Grundlayout mithilfe des Goldenen Schnitt

Was auffällt ist, das Gegenüber der Scribble Skizze, das die beiden Inhaltsbereiche die Seiten getauscht haben. Grund der Änderung ist die allgemeine Platzierung einer Navigation, diese wird grundsätzlich im oberen Bereich oder im linken Bereich der Webseite gefunden. Welches ein schnelleres Zurechtfinden der Nutzer, durch persönliche Erfahrungen im Web, bietet.

„Die traditionelle Hauptnavigation befindet sich dann meist links, wo sie mittlerweile von über 50 Prozent der Besucher erwartet wird.“[1](1)

Im oberen Bereich des linken Inhaltsbereiches ist das Logo der Handspiel GmbH zu sehen, plus die Überschrift für den Usability-Tag, danach folgt die Navigation. Weil noch genügend Platz im linken Inhaltsbereich unter der Navigation, zur Verfügung stand, wurde nach Absprache mit dem Auftraggeber noch eine Box für Tipps und Infos hinzugefügt.

Im rechten Inhaltsbereich findet sich im oberen Bereich eine Headline, die den Titel der Seite und die aktuelle Zeit anzeigt. Allgemein wurde darauf geachtet das die Abstände jeweils ein Vielfaches von 12 betragen. Weitere Informationen zu dem rechten Inhaltsbereich, sind in den Kapitel 3.1.2.3 zu finden. Das gesamte Layout, der digitalen Umsetzung, befindet sich im Anhang B.

3.1.2.2 Navigation

Abb. 3.1-3 Navigation mit aktiven Listenelementen

[1] (Fischer, 2009), s.563.

In der Navigation werden drei Zustände unterschieden, aktiv, hover und nicht aktiv. Die grafische Gestaltung und deren Effekte sind bei den Hover und Aktiv Zuständen dieselben. Folgende Gestaltung wurde für die einzelnen Zustände vorgenommen:

- Aktiver und Hover Zustand: Menschen werden nicht Schwarzweiß dargestellt und den Buttons wurde der Tiefenwirkungseffekt durch den Schatten entfernt. Zusätzlich ziert ein leichter Schein den oberen Teil des Hintergrundes. Siehe Abb. 3.1-3
- Nicht aktiv: Menschen sind in Schwarzweiß dargestellt und verfügen über einen Schatten. Siehe Abb. 3.1-4(r. B.)

Abb. 3.1-4 Zustände der Navigationsbuttons

3.1.2.3 Rechter Inhaltsbereich

Startseite bzw. Beitragsseite

Farblich dadurch gekennzeichnet, dass die Umrandung Rot dargestellt ist. Da nicht alle Vorträge auf die Seite passen, werden immer nur vier Einträge dargestellt, durch einen Trennstrich werden diese separiert.

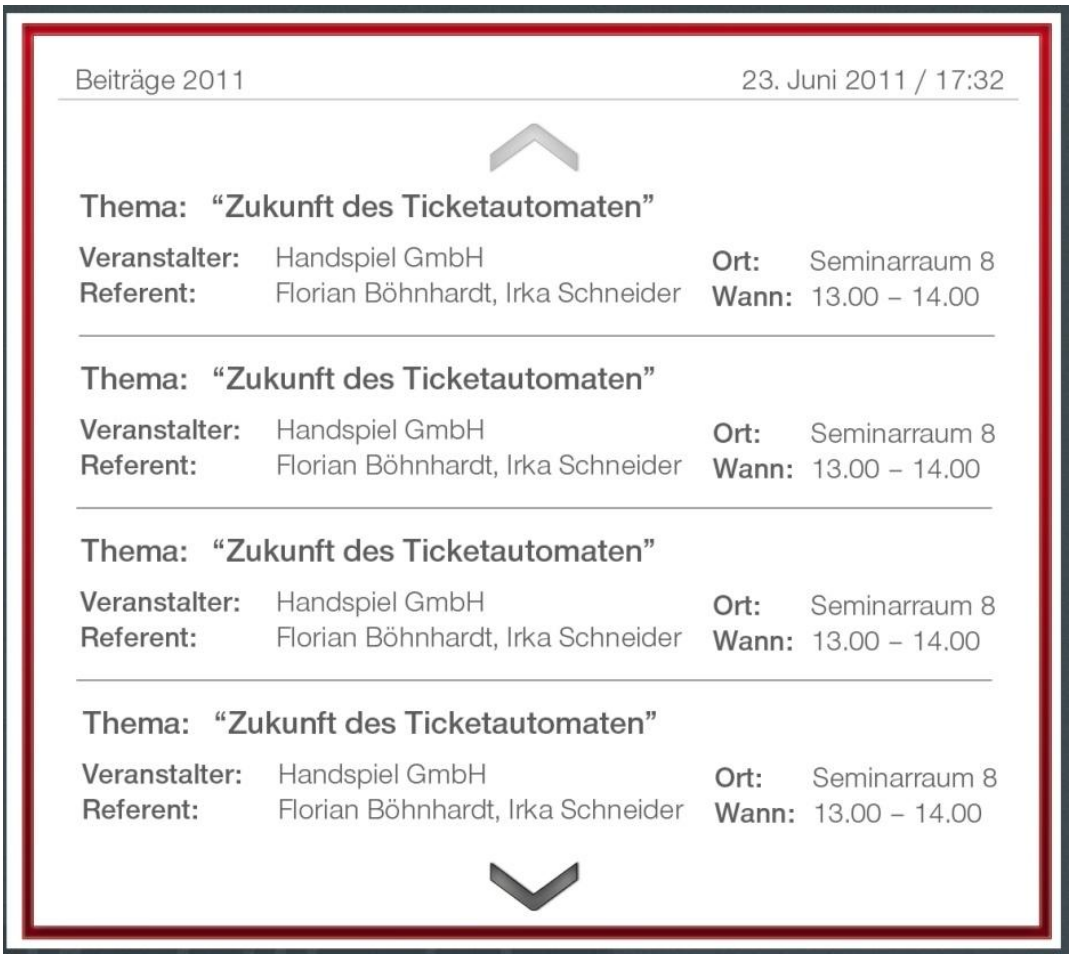


Abb. 3.1-5 Ansicht Layout-Entwurf Beitragsseite

Wenn die Einträge größer als vier werden, soll es über Navigation möglich sein, mittels der grauen Pfeile oben und unten, die nächsten oder vorigen Einträge anzuzeigen. Sobald keine vorige Seite oder folge Seite vorhanden ist, wird der Pfeil farblich aufgehellt und die Deckkraft reduziert.

Beispielseite

Optisch ist die Beispielseite durch einen blauen Rahmen gekennzeichnet. Im linken Teil des Inhaltsbereiches befindet sich die Navigation. Der Umfang der dargestellten Inhalte, in der Navigation wurde auf fünf beschränkt. Sobald ein Miniaturbild ausgewählt wurde, soll dieses im rechten Teil, als Detailbild erscheinen. Zusätzlich soll der Titel des Bildes und eine Beschreibung, angezeigt werden. Nach der Auswahl wird das aktive Miniaturbild, mit einem Schatten und einer Kontur versehen, alle nicht aktiven Elemente, werden durch Reduzierung der Deckkraft gekennzeichnet. Zur optischen Verschönerung wurde das Detailbild auch mit einem Schatten und einer Kontur versehen.

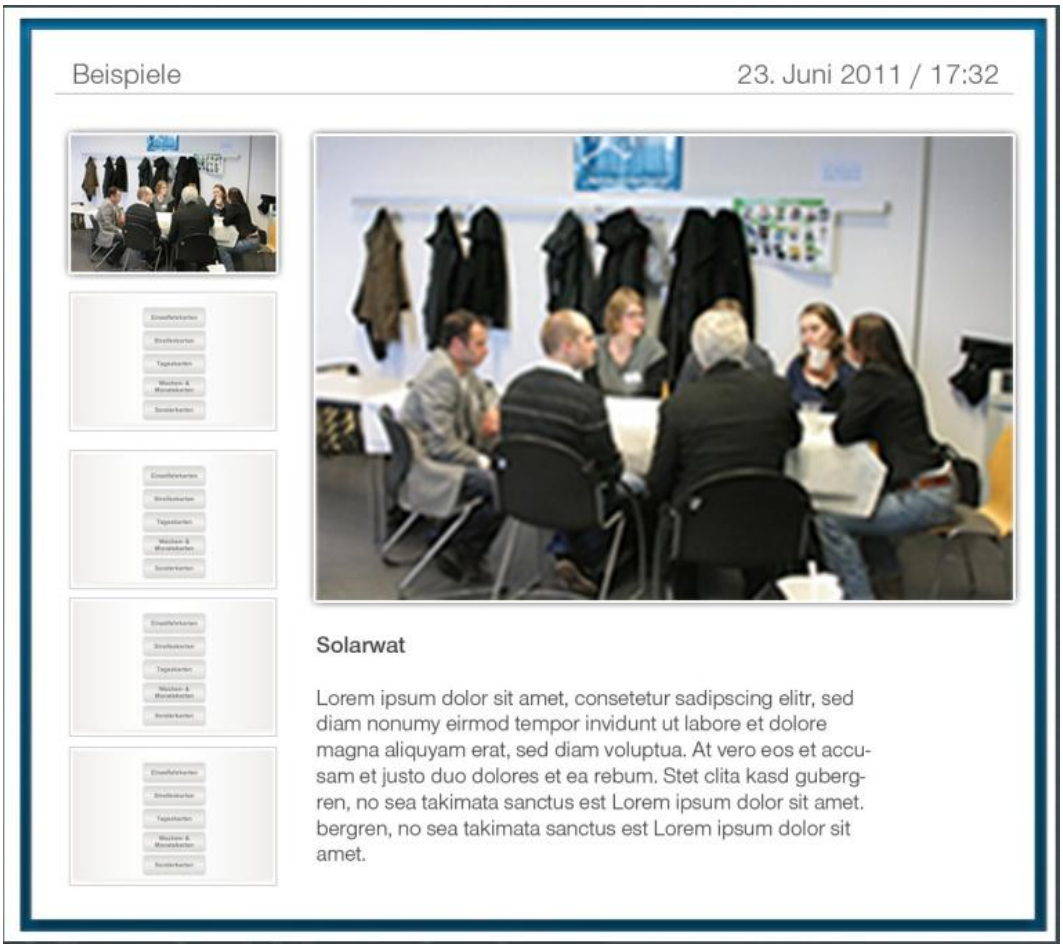


Abb. 3.1-6 Ansicht Layout-Entwurf Beispielseite

Videoseite

Bei der Videoseite wird der Rahmen in einen grünen Farbton dargestellt. Im mittleren Teil des Inhaltsbereiches, ist der Videoplayer und die zugehörige Steuerung zusehen. Darunter befindet sich die Navigation, die für den Wechsel zwischen den Videos zuständig ist. Der Wechsel soll, nach der Auswahl eines der Vorschaubilder erfolgen. Wenn mehr Videos als drei Stück vorhanden sind, soll mittels der Pfeile, der Wechsel zu den nächsten Videos ermöglicht werden. Das aktuelle Video soll wie bei der Beispielseite, durch einen Schatten gekennzeichnet sein. Zusätzlich wird bei den nicht aktiven Vorschaubildern die Deckkraft reduziert.

Abb. 3.1-7 Ansicht Layout-Entwurf Videoseite

Sponsorenseite

Auf dieser Seite werden die beteiligten Sponsoren dargestellt. Das Merkmal der Sponsorenseite, ist der graue Rahmen, der sich um den Inhaltsbereich befindet. Der Aufbau wurde so festgelegt, dass das Logo links neben den Adressdaten angeordnet ist und die maximale Anzahl der Sponsoren, wurde auf vier festgelegt.

Abb. 3.1-8 Ansicht Layout-Entwurf Sponsorenseite

3.1.2.4 Hintergrundgestaltung

Hier wurde der gleiche Hintergrund, wie auf der Handspiel-Webseite[1] gewählt. Dieser beinhaltet eine Struktur und wurde farblich mit folgenden #5C5C5C-Farbton hinterlegt. Zusätzlich befindet sich am Fuße der Webanwendung. eine Grafik, die aufsteigende Pixel darstellt, siehe Abb. 3.1.9.

Abb. 3.1-9 Hintergrund der Webanwendung

Während der Entwicklung kann es zu gestalterischen Abweichungen kommen, da bestimmte Inhalte noch nicht vorhanden bzw. festgelegt wurden und somit eine 1:1-Umsetzung der erstellten Layouts nicht gegeben ist. Nachdem alle grafischen Inhalte, durch den Auftraggeber freigegeben wurden, folgt im nächsten Kapitel die Umsetzung der Anwendung.

[1] http://www.handspiel.net/

3.2 Implementierung

3.2.1 Einrichtung Content Management-System

3.2.1.1 Installation der Testumgebung

Als Erstes muss der RedaxoWinstaller heruntergeladen werden, dieser ist auf der RedaxoWinstaller Webseite[1] zu finden. Nach dem Download muss die Installationsdatei ausgeführt werden. Die Installation verhält sich wie jede andere Installation, deswegen ist keine Erklärung notwendig. Im Installationspaket enthalten, ein Apache-Server mit der Datenbank MySQL und den Scriptsprachen PHP. Auch die aktuelle Version 4.3.2 von Redaxo ist in diesem Paket enthalten.

Falls keine Änderungen bei der Installation vorgenommen wurden, findet sich nun auf den Desktop eine Startdatei. Nachdem Ausführen dieser Datei, öffnet sich ein Kontrollfenster, welches zum Starten, Stoppen und Beenden der Apache und MySQL Server verwendet wird. (Abb. 3.2-1)

Abb. 3.2-1 Kontrollfenster der Testumgebung

Nach dem Betätigen des „Redaxo starten“ Buttons, wird die Informationsseite, im festgelegten Standardbrowser geöffnet. (Abb. 3.2-2) Auf dieser Seite befinden Informationen, die für eine Umsetzung einer Webanwendung notwendig sind. Folgende Informationen können abgerufen werden, REDAXO-Instanzen, Hinweise zu Links, Tools und der Dokumentationen.

[1] http://redaxowinstaller.de/downloads.html

Im Paket enthalten sind zwei REDAXO-Instanzen, die zur Weiterverarbeitung verwendet werden dürfen. Einmal SimpleDemo, wo das REDAXO schon viele sinnvolle Add-ons, Module, Templates installiert hat. Die andere Instanz BlankSite, bietet hingegen nur installierte Add-ons, aber keine vorgefertigten Module oder Templates. Das Frontend und Backend, der beiden Instanzen ist über Informationsseite erreichbar.

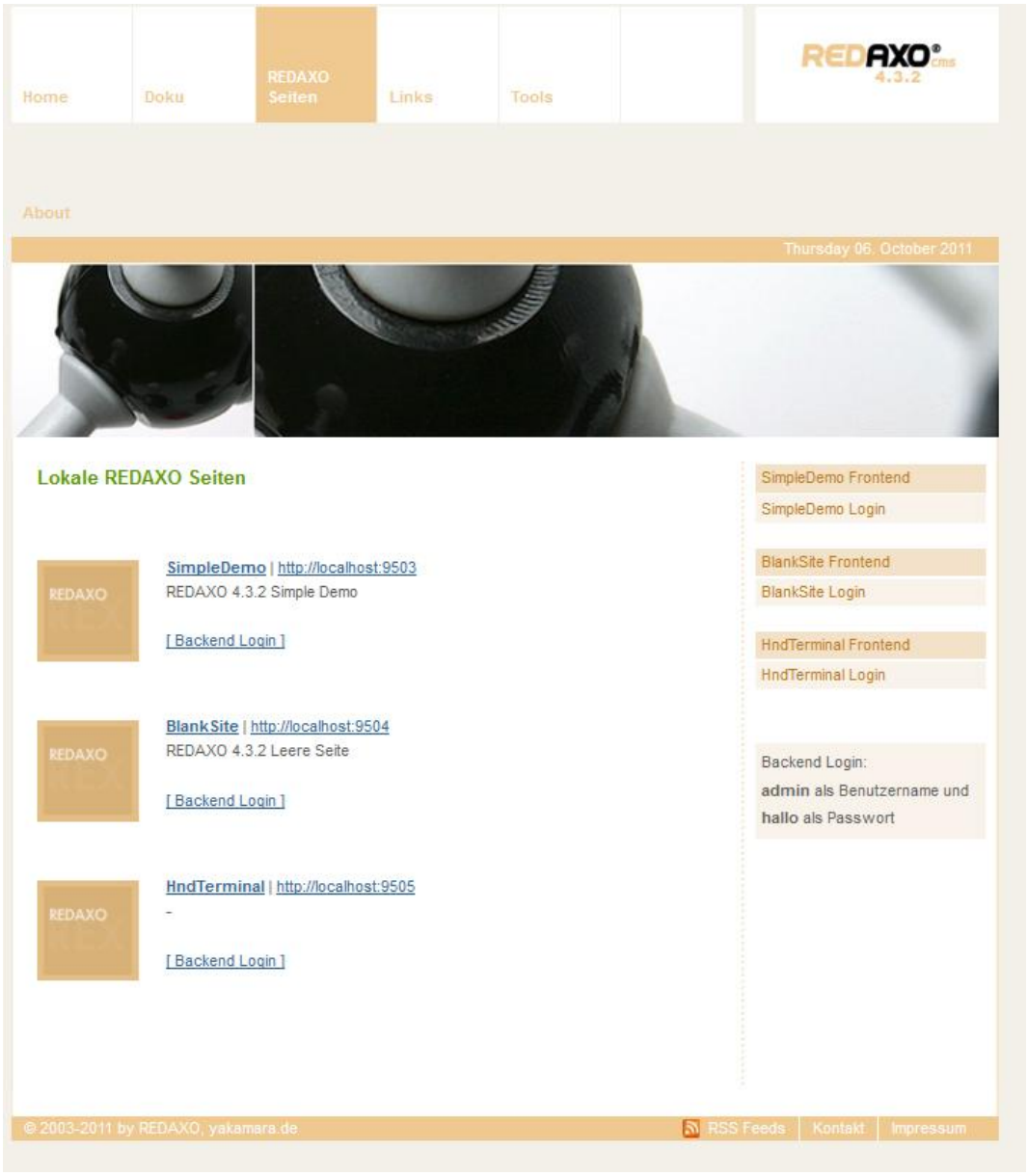


Abb. 3.2-2 Informationsseite RedaxoWinstaller

Auch das Hinzufügen neuer Instanzen ist möglich. Im Kontrollfenster(Abb. 3.2-1) muss dafür der Menüpunkt Optionen ausgewählt werden. Im Reiter Redaxo Instanzen ist es dann möglich, die aktuellen Instanzen zu

bearbeiten oder zu löschen. Neue Instanzen können auch eingefügt werden, siehe Abb. 3.2-3.

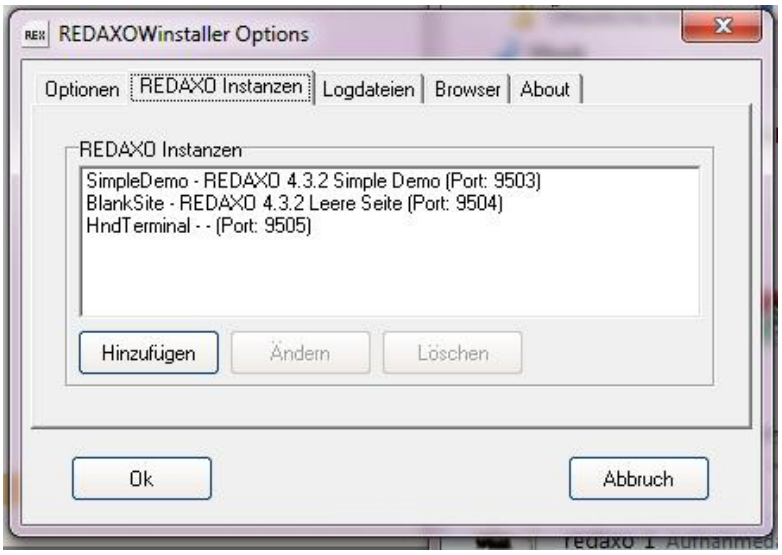


Abb. 3.2-3 RedaxoWinstaller Optionen, Reiter REDAXO Instanzen

3.2.1.2 Ordnerstruktur Webanwendung

Die Ordnerstruktur wurde so aufgebaut, dass eine klare Trennung zwischen den hinzugefügten Dateien und CMS Dateien gegeben ist. So werden in dem Ordner files, alle Dateien, die zur Veränderung des Frontends notwendig sind, abgespeichert. Aber auch multimediale Inhalte sind in diesen Ordner zu finden. Alle Dateien, die für das CMS notwendig sind, befinden sich im Ordner Redaxo.

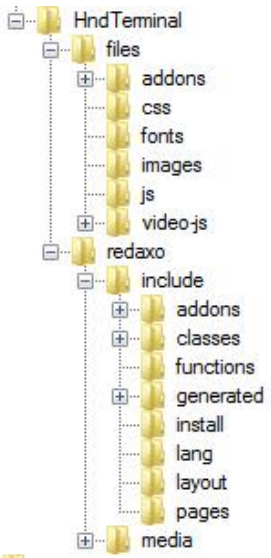


Abb. 3.2-4 Ordnerstruktur Webanwendung

3.2.1.3 Import & Export

Um Daten zu Exportieren, gibt es zwei Wege:

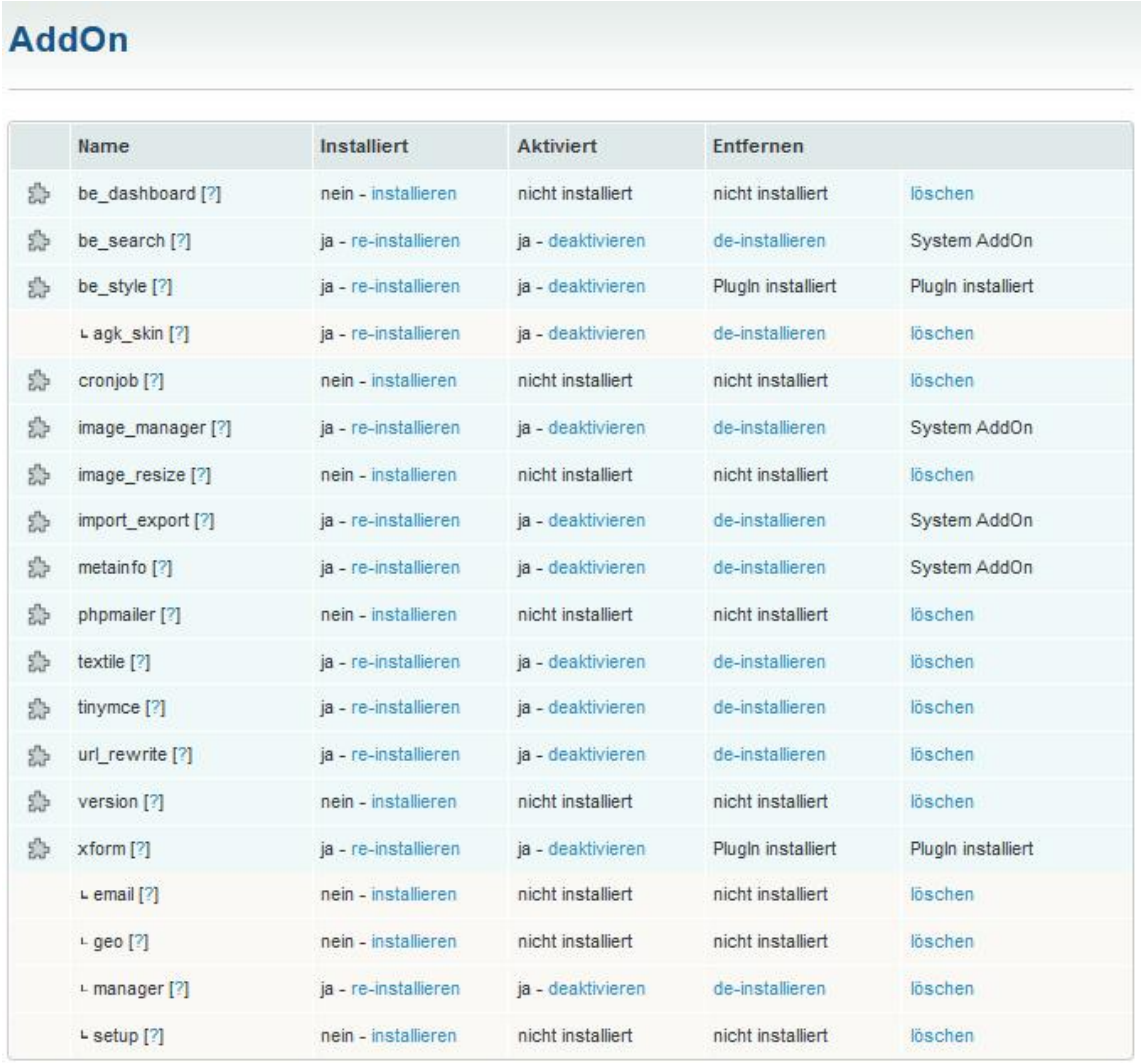
Der erste Weg ist über das integrierte Im-/Export Add-on, welches ermöglicht die Datenbank und Dateien der Anwendung zu exportieren. Sobald aber die Größe der beinhalteten Files über 300 MB geht, wird eine Fehlermeldung ausgegeben. Geeignet ist diese Funktion, nur für kleine Webseiten.

Der zweite Weg führt über die Sicherung mittels FTP, hierfür wird ein FTP-Programm z.B. Filezilla[1]. Der Export der Datenbank muss mittels phpMyAdmin erfolgen, die lokale Datenbank die über den RedaxoWinstaller erstellt wurde, ist auf der Informationsseite (Abb. 3.2-2) unter Tools zu finden. Bei einem Webserver ist phpMyAdmin über das Hostingbackend zu erreichen. Wenn sich der Datenbankname und die zugehörigen Zugangsdaten geändert haben, müssen diese Angaben, in der master.inc.php geändert werden. Die Datei ist im Redaxo-Ordner unter include zu finden, siehe Abb. 3.2-4.

3.2.1.4 Verwendete Add-ons

Damit Add-ons[2] verwendet werden können, müssen diese im Ordnerverzeichnis redaxo\include\addons gespeichert werden siehe Abb. 3.2-4. Danach muss eine Installation und Aktivierung erfolgen, dies kann im Backend unter Add-ons ausgeführt werden. In der nachfolgenden Abb. 3.2-5, ist die Anzahl der installierten Add-ons zu erkennen. Um mehr Informationen über die einzelnen Add-ons zu erfahren, muss am Ende des Namens, das Fragezeichen angeklickt werden.

[1] http://www.filezilla.de/
[2] Add-on – Erweiterung der Software durch zusätzliche Funktionen



AddOn

	Name	Installiert	Aktiviert	Entfernen	
	be_dashboard [?]	nein - installieren	nicht installiert	nicht installiert	löschen
	be_search [?]	ja - re-installieren	ja - deaktivieren	de-installieren	System AddOn
	be_style [?]	ja - re-installieren	ja - deaktivieren	PlugIn installiert	PlugIn installiert
	↳ agk_skin [?]	ja - re-installieren	ja - deaktivieren	de-installieren	löschen
	cronjob [?]	nein - installieren	nicht installiert	nicht installiert	löschen
	image_manager [?]	ja - re-installieren	ja - deaktivieren	de-installieren	System AddOn
	image_resize [?]	nein - installieren	nicht installiert	nicht installiert	löschen
	import_export [?]	ja - re-installieren	ja - deaktivieren	de-installieren	System AddOn
	metainfo [?]	ja - re-installieren	ja - deaktivieren	de-installieren	System AddOn
	phpmailer [?]	nein - installieren	nicht installiert	nicht installiert	löschen
	textile [?]	ja - re-installieren	ja - deaktivieren	de-installieren	löschen
	tinymce [?]	ja - re-installieren	ja - deaktivieren	de-installieren	löschen
	url_rewrite [?]	ja - re-installieren	ja - deaktivieren	de-installieren	löschen
	version [?]	nein - installieren	nicht installiert	nicht installiert	löschen
	xform [?]	ja - re-installieren	ja - deaktivieren	PlugIn installiert	PlugIn installiert
	↳ email [?]	nein - installieren	nicht installiert	nicht installiert	löschen
	↳ geo [?]	nein - installieren	nicht installiert	nicht installiert	löschen
	↳ manager [?]	ja - re-installieren	ja - deaktivieren	de-installieren	löschen
	↳ setup [?]	nein - installieren	nicht installiert	nicht installiert	löschen

Abb. 3.2-5 Redaxo Add-ons

XForm

Hervorzuheben ist das Add-on XForm, umgesetzt von der Yakamara Media GmbH & Co. KG, ermöglicht es die Definition eines Formulars und die Einstellungen verschiedener Validierungsmöglichkeiten. Auch eigene Aktionen sind definierbar, z.B. Einträge in einer Datenbank erstellen und E-Mails verschicken.

Das XForm Add-on wird in dieser Anwendung dazu verwendet, um Inhalte mittels Formularfelder zu erstellen. Damit Redakteure die Formularfelder nutzen können, müsse diese vorab durch einen Admin angelegt werden. Um Formularfelder zu erstellen, muss der Table Manager im Backend unter den Menüpunkt XForm ausgewählt werden. Im Table Manager

werden alle vorhandenen Tabellen, in denen sich die Formularfelder befinden, aufgelistet. Nun können neue Tabellen hinzugefügt werden, vorhandene Tabellen gelöscht oder der Tabellenfelder bearbeitet werden. Bei der Bearbeitung der Tabelle, werden alle bereits erstellten Formularfelder angezeigt. Formularfelder können dann, bearbeitet, gelöscht oder neu angelegt werden. Bei der Anlegung eines Formularfeldes wird eine Auswahl von verschiedenen Feldern zur Verfügung gestellt. Es wird zwischen Value-Feldern und Validate-Feldern unterschieden. Die Value-Felder erzeugen Eingaben, danach können diese mit den Validate-Felder überprüft werden. Alle erzeugten Tabellen, die über die XForm erstellt wurden, werden in derselben Datenbank, in der sich auch die Redaxo-Tabellen befinden, abgespeichert.

Image Manager

Dieses Add-on wird dazu verwendet um Grafiken, mit Effekten zu versehen. So können Grafiken zum Beispiel mit einer Unschärfe oder Schärfung versehen werden. Farblich können die Grafiken zum Beispiel durch einen Schwarz Weiß oder Sepia Filter geändert werden. In unserer Webanwendung wird der Effekt „resize“ verwendet, dieser skaliert die verwendeten Bilder auf passende Größe. Zu finden ist der Image Manager in der Navigationsleiste im Backend unter Add-ons. Die Einbindung erfolgt über eine Notierung in der URL.

Beispiel:

```
index.php?rex_img_type=ImgTypeName&rex_img_file=ImageFileName
```

Wenn das Content Management-System installiert ist und alle notwendigen, Add-ons eingebunden, installiert und aktiviert sind. Ist der nächste Schritt, die Module und Templates, die zur Darstellung im Frontend notwendig sind, zu erzeugen.

3.2.2 Templates

Um eine bessere Übersicht über den Programmiercode zu erhalten, wurden mehrere Templates erstellt. So gibt es folgende Templates:

- Default
- Content-Right
- Html-head
- Navigation:Links

Der gesamte Quellcode für die Templates ist im Anhang A zu finden.

3.2.2.1 Default-Template

Das Default-Template könnte auch als Herzstück, bezeichnet werden, denn das Default-Template bestimmt, wie der allgemeine Aufbau der einzelnen Inhaltsseiten im Frontend aussieht, siehe Abb. 3.2-6.

```
REX_TEMPLATE[3]
<body>
    <div id="content">
        <div id="leftContent">
            <div class="leftContentHead">
                <div class="logo"></div>
                <header>
                    <h1>Usability Tag 11.11.2011</h1>
                </header>
            </div>
            <nav class="leftNavigation">
            REX_TEMPLATE[2]
            </nav>
            <article class="infoBox">
            REX_ARTICLE[ctype="2"]
            </article>
            <article class="tipBox">
            REX_ARTICLE[ctype="1"]
            </article>
        </div>
        <div id="rightContent">
        REX_TEMPLATE[4]
        </div>
    </div>
    <div id="backgroundImage"></div>
</body>
</html>
```

Abb. 3.2-6 Default-Template

Da aber das Default-Template, ohne die anderen Templates und Artikel, den Inhaltsbereich nicht korrekt darstellt, müssen alle notwendigen Artikel und Templates eingebunden werden. Die Einbindung eines Template erfolgt über `REX_TEMPLATE[id]` und über `REX_ARTICLE[]`, erfolgt die Einbindung der Artikel, in denen auch die einzelnen Module integriert sind. Damit nicht alle Artikelspalten des Artikels ausgegeben werden, wird mittels `ctype`, die gewünschte Spalte ausgewählt.

3.2.2.2 Content-Right-Template

Im Content-Right-Template, befinden sich alle Elemente, die zur Gestaltung und Einbindung des rechten Inhaltsbereichs notwendig sind. Über ctype werden die unterschiedlichen Ausgabe Module eingebunden. In diesem Template, befinden sich zusätzliche PHP und JavaScript-Funktionen, die dafür zuständig sind, den korrekten Seitennamen und die aktuelle Uhrzeit, im `<header>` wiederzugeben, siehe dazu auch Kapitel 3.2.5.

```
<div class="borderColorContent red">
    <div id="contentArea">
        <header>
            <h2>Beiträge 2011 </h2>
            <time class="hndTime">15:46</time>
        </header>
        <section id="mainContent">
        REX_ARTICLE[ctype="3"]
        </section>
    </div>
</div>
```

Abb. 3.2-7 Content-Right-Template

3.2.2.3 HTML-head-Template

Im HTML-head-Template, sind alle Einbindungen zu finden, die für die korrekte Darstellung und Funktionalitäten der Webanwendung, verantwortlich sind, dazu zählen alle CSS Dateien und JavaScript Dateien. Auch Metatags sind in dem Headbereich eingebunden, diese werde je

nach Eingabe, für jede Seite unterschiedlich erstellt. Da aber keine Suchmaschinenoptimierung benötigt wird, sind diese Tags nicht gefüllt.

Damit ein schnelles Hinzufügen und Entfernen von Einbindungen gegeben ist, wurde der gesamte Code, in einem anderen Template abgespeichert. Genauere Informationen zum HTML-head-Template, sind im Anhang A zu finden.

3.2.2.4 Navigation: Links

Dieses Template ist dafür zuständig, eine komplette Navigation zu erstellen. Damit bekannt ist, wie viel Menüpunkte dargestellt werden, wird mittels RexScript, die einzelnen Kategorien ausgelesen, die sich im Strukturverzeichnis befinden. Danach wird mittels PHP, jedem Listenelement eine Klasse über die ID zugeordnet. Hierfür wurde vorher ein Array erstellt, wo die Farbzuweisen, für die einzelnen Buttons zu finden sind. Zusätzlich wird der aktuellen Kategorie, die Klasse aktiv zugewiesen. Siehe auch Anhang A.

Die Ausgabe der Navigation im Frontend sieht dann so aus:

```
<ul class="nav1">
    <li class="redbutton">
        <div class="naviIcon"></div>
        <a href="index.php?article_id=1">Beiträge 2011</a>
    </li>
    <li class="bluebutton">
    <li class="greenbutton">
    <li class="greybutton active">
</ul>
```

Abb. 3.2-8 HTML Ausgabe Navigation Frontend, Sponsorenseite

3.2.3 Module

Module dienen dazu, Redakteuren die Einbindung von Inhalten zu erleichtern und die Ausgabe der Inhalte im Frontend zu steuern. Da zur Inhaltseinbindung das XForm Add-on (siehe Kapitel 3.2.7)verwendet wird, muss hierfür, nur eine Ausgabe über die Module erfolgen. Eine Ausnahme gibt es und das ist die Einbindung von Texten in die Infobox. Im folgenden Beispiel wird gezeigt, wie die Ausgabe der XForm realisiert wird.

Die gesamte Ausgabe basiert auf PHP deswegen muss der gesamte Code integriert werden, dies erfolgt über `<?php Code ?>`.

Das folgende Beispiel zeigt die Modulausgabe für die Bildergalerie. Damit die Inhalte der Bildergalerie verwendbar sind, muss zuerst auf die Bildergalerie Tabelle, in der Datenbank zugegriffen werden.

```
$db_table = "bildergalerie";

$sql = new rex_sql;

$sql->debugsql = 0;

$sql->setQuery("SELECT * FROM $db_table");
```

In den anderen Modulen funktioniert dies auf die gleich Weise, nur das sich die Datenbank Tabellen unterscheiden. Danach wird jede Zeile einzeln ausgelesen. Da nur fünf Beispiele in der Bildergalerievorschau dargestellt werden, müssen nur die ersten fünf Zeilen ausgelesen werden.

```
for($i=0;$i<=4;$i++){

…

}
```

Jetzt müssen nur noch die einzelnen Werte, aus der gewünschten Spalte gelesen werden. Dies erfolgt über:

```
echo $sql->getValue("mediendatei");

echo $sql->getValue("imageName");
```

Am Schluss muss noch ein Counter hinzugefügt werden, der dafür zuständig ist, dass die nächste Zeile ausgewählt wird. Das Erhöhen endet, wenn alle fünf Zeilen ausgelesen sind.

```
$sql->counter++;
```

Um die Inhalte besser mit CSS formatieren zu können, werden diese in Elemente eingeschlossen, mehr Informationen dazu finden sich im Anhang A, in diesem befinden sich auch alle Quellcodes der Module.

Modul Beitragsseiten

Bis auf die Beitragsseiten erfolgt die Ausgabe aller XForm Module identisch. Auf der Beitragsseite müssen folgende Anpassungen durchgeführt werden.

Da die Beiträge unterschiedliche Events enthalten, die jeweils unterschiedliche Spalten Inhalte ausgeben, musste hierfür eine Regelung bei der Ausgabe erfolgen. Dies erfolgte über eine if- Anweisung und einer Kombination von `if` und `else`.

```
if($event =="Pause")
     {.. Inhalt .. }
elseif ($event == "Beitrag")
     { …Inhalt… }
elseif($event == "Einleitung" || $event == "Abschluss")
     { … Inhalt… }
```

Des Weiteren wurden die Einträge, in Eintragsseiten eingeteilt, die jeweils vier Einträgen beinhalten. Und so funktioniert es:

```
$z = $i%4 ;
```

Damit bekannt ist, an welchen Stellen, sich der Eintragsseitenanfang und Ende befindet. Wird mit den Modulo Operator, die Seiten jeweils in 4 Einträge geteilt. Denn dieser gibt immer der Rest von 4 zurück, in diesem Fall immer den Rest 0-3. Diese Zahlen können als Orientierung, für das

einfügen von Anfangs- und Endtags, verwendet werden. So steht die 0 für den Seitenanfang und die 3 für das Ende.

```
if ( $z == 0) {

  echo '<div class="entryPage">';

}

else {}
```

Zusätzlich musste noch eine Ausnahme Regel für Eintragsseiten die weniger als vier Einträge beinhalten, festgelegt werden. Hierfür wird nach dem Auslesen der letzten Zeile, das Ende eingefügt.

```
$y = $sql->getRows();

if ($z == 3 || $i == $y-1) {

  echo '</div>';

}

else {}
```

Modul Videogalerie

Zur Wiedergabe von Videos wird der VideoJS HTML 5 Player verwendet. Der Player ist eine freie und Open Source gestützte Software. Dessen Aussehen mittels CSS, angepasst werden kann. Vorgefertigte Skins sind auf der auf der Webseite VideoJS[1] erhältlich. Zur Anwendung wird nur eine CSS-Datei und eine Klassenzuweisung in einem VideoJs-Element, benötigt.

Funktionen wie Vollbildansicht und Lautstärkeregelung sind im Packet enthalten. Auch das gleiche Erscheinungsbild in verschiedenen Browser wird gewährleistet. Sobald der HTML5 Player nicht unterstützt wird, ist eine Absicherung durch einen Flashplayer gegeben.

[1] http://videojs.com/

Einbindung

Damit der Player einwandfrei funktioniert, wird die video.js und die video-js.css Datei benötigt. Die Einbindung erfolgt über den HTML-head.

```
<head>
...
  <script src="video.js" type="text/javascript" charset="utf-8"></script>
  <link rel="stylesheet"
      href="video-js.css"
      type="text/css"
      media="screen"
      title="Video JS"
      charset="utf-8" >
...
</head>
```

Abb. 3.2-9 Einbindung VideoJS-Dateien in den HTML-head

Damit der Player im Frontend sichtbar ist, muss folgender Code in der Modulausgabe stehen.

```
 <!-- Begin VideoJS -->
<div class="video-js-box">
  <video class="video-js" width='720' height='540' controls preload>
    <source src="" type='video/ogg; codecs="theora, vorbis"' />
  </video>
</div>
<!-- End VideoJS -->
```

Abb. 3.2-10 Moduleinbindung VideoJS

3.2.4 Elementanpassungen mit CSS

Um einen Überblick über die Anpassungen mittels CSS zu erhalten, wurden diese in mehrere Dateien aufgeteilt. Wodurch das Erweitern und Anpassen von Attributen, an Elementen erleichtert wird. Die Einbindung der CSS-Dateien geschieht im HTML-head-Template. Die gesamten CSS-Dateien sind im Ordner files/css zu finden. Der gesamte Quellcode zu den CSS-Dateien ist im Anhang A zu finden. Folgende CSS-Dateien wurden erstellt:

reset.css

Die reset.css wird benötigt, um die vordefinierten Eigenschaften der Browser auf einen gemeinsamen Nenner zu bringen. So werden Elemente und deren Eigenschaften, wie z.B. margin auf null gesetzt.

navigation.css

Navigation.css ist dafür zuständig, dass Elemente der Navigation, die richtigen Grafiken und Eigenschaften zugeordnet bekommen. Eigenschaften, wie zum Beispiel für die Platzierung oder Größe der Buttons. Die Grafikzuweisung erfolgt über Sprites[1] und dem Backgroundpositon-Attribut.

main.css

In der main.css befindet sich primär die Information für die Positionierung, Breite und Längenangaben sowie die Art der Positionierung, größtenteils für div´s aber auch für andere Elemente.

content.css

Hier finden sich primär, alle Attribute, die zur Gestaltung von Elementen beitragen. So werden zum Beispiel Elemente mit Schatten versehen, die mittels CSS3 erstellt werden.

font.css

In der Font.css sind Schrifteinbindungen sowie die Veränderungen von Schriften zu finden. Wie zum Beispiel: Schriftgröße. Schriftfarbe, usw.. Auch das Allgemeine zuweisen von Attributen an Elementen, wird hier geregelt.

[1] Sprite – Sammlung von Grafiken in einer Datei.

3.2.5 Funktionen

3.2.5.1 JavaScript

Um eine aktuelle Zeit, im Inhaltsbereich darzustellen, wird die Funktion `aktuelleZeit()` benötigt. Diese befindet sich in der JavaScript-Datei zeit.js.

Zuerst wird die aktuelle Zeit benötigt, dafür wird `Date()` verwendet.

```
jetzt = new Date();
```

Da jetzt die aktuelle Zeit bekannt ist, müssen notwendige Daten ausgelesen werden.

```
stunde = jetzt.getHours();
sekunde = jetzt.getSeconds();
```

Danach werden die Daten zur richtigen Darstellung angepasst. So wird jeder Zahl die kleiner ist als 10, eine 0 voran gestellt. Die Überprüfung muss jeweils für die Stunden und Minuten durchgeführt werden.

```
if ( jetzt.getHours() < 10) {

     stunde = "0" + jetzt.getHours();

 }else { stunde = jetzt.getHours();}

if ( jetzt.getMinutes() < 10) {

     minute = "0" + jetzt.getMinutes();

}else {minute = jetzt.getMinutes();}
```

Mit den jetzigen Daten, kann eine Zusammenstellung der gewünschten Zeit erfolgen. Danach wird mit `return` die Zeit zurück gegeben werden.

```
zeit = stunde + ":" + minute;

return zeit;
```

Um eine ständige Aktualisierung der Zeit zu gewährleisten, wird mittels `window.setInterval`, die Funktion aller 1s aktualisiert.

```
window.setInterval(function() {
   $(".hndTime").html(aktuelleZeit());
   },
   1000
);
```

3.2.5.2 jQuery Effekte

Damit jQuery einwandfrei funktioniert, wird die jQuery.js Datei, in den HTML-Head eingebunden. Damit keine Komplikationen auftreten, wird immer die aktuellste Version verwendet, Stand der jQuery-Version bei der Entwicklung war 1.6.1. Erhältlich ist die Datei auf der Webseite von jQuery.[1]

In den nächsten Abschnitten werden Quellcodeausschnitte zur Erklärung der einzelnen Funktionen verwendet. Der gesamte Quellcode ist im Anhang A zu finden.

Bildergalerie und Videogalerie

Für Bilder- und Videogalerie, müssen Funktionen erstellt werden, die ein Wechsel zwischen verschiedenen Quellen ermöglicht und die aktuelle Auswahl darstellt. Folgende zwei Funktionen sind dafür zuständig. Für die Bildergalerie `changePicture()`, zu finden in der bildergalerie.js und für den Videowechsel `changeVideo()`, zu finden in der videogalerie.js.

Als Erstes muss festgelegt werden, welche Beschreibung und welches Bild beim ersten Aufruf und neu Laden der Seite, dargestellt werden soll. Dafür wird eine Zuweisung einer Aktiv-Klasse, an das erste Element benötigt, in diesem Fall, für die Bildergalerie die Klasse `current` und bei den Videos die Klasse `active`. Damit kann das aktuelle Element gezielt manipuliert werden.

[1] http://jquery.com/

Im Script sieht das so aus:

```
$('.smallImage:first-child').addClass('current');

$(".videosEntry:first").addClass("active");
```

Nachdem dies festgelegt wurde, kann man sich die Informationen für die Hauptansicht holen, die aus dem Detailbild und deren Beschreibung besteht.

Für die Bildergalerie:

```
$('.bigImage img')

     .attr( 'src' ,

          $('.smallImage.current img').attr('src')
          );

$(".imageTitle h3")

     .html($('.smallImage.current
img').attr('title'));

$(".imageDescription p")

     .html($(".smallImage.current p").text());
```

Bei den Videos so:

```
$(".video-js source")

 .attr( 'src' , "./files/" +
$(".videosEntry.active.videoFormat").html());
```

Beim Anklicken eines der Miniaturbilder, aus der Navigation, sollen die zugehörigen Details in der Hauptansicht angezeigt werden. Funktionieren tut dies mittels des Events `click`, welches eine Reihe von Ereignissen auslöst.

```
smallImage.click( function(){

… Ereignisse … })
```

Damit eine erneute Auswahl verhindert wird, muss überprüft werden, ob das Aktiv-Element ausgewählt ist. Sobald dies der Fall ist, werden keine Events ausgegeben. Da sich die beiden Funktionen ähneln, wird die Erklärung nur für die Bildergalerie gezeigt. Sieht folgendermaßen aus:

```
if($(this).is(".current")){
     return false;
 }
else { …Ereignisse… }
```

Jetzt werden noch die Ereignisse benötig. Zuerst wird das Aktiv-Element gewechselt, damit nicht zwei Aktiv-Elemente vorhanden sind, muss vorher die Zuweisung auf dem aktuellen Element entfernt werden. Danach wird dem ausgewählten Element, die Klasse `current` zugewiesen.

```
$('.smallImage.current')
     .removeClass('current');
$(this)
     .addClass('current');
```

Als Nächstes werden die aktuellen Detailansicht–Elemente mit `fadeOut` ausgeblendet danach wird ein Callback[1] ausgeführt. Der sorgt dafür, dass die Detail-Elemente mit den aktiven Daten gefüllt werden. Danach wird mittels `fadeIn` die Elemente wieder eingeblendet.

```
$(".imageDescription p")
     .fadeOut( 1000 , function () {
          $(this)
          .html($(".smallImage.current p").text())
          .fadeIn(1000);
       });
```

Bei dem Detailbild funktioniert es fast identisch. Nur das die aktuellen Attribute des Hauptbildes entfernt werden und mit den Attributen des Vorschaubildes gefüllt werden.

[1] Rückruf der nach dem Beenden der Hauptfunktion ausgeführt wird

```
$('.bigImage img')
            .fadeOut( 1000 , function () {
              $(this)
          .removeAttr('src' )
          .attr(  'src'  ,  $('.smallImage.current
          img').attr('src'))
          .fadeIn(1000); });
```

Der gleiche Ablauf wird auch bei dem Videogalerie-Funktionen verwendet, es werden Klassen hinzugefügt und entfernt. Attribute werden entfernt und mit den aktuellen Werten gefüllt. Damit der Player sich im Pause Zustand befindet, wenn ein Videowechsel ausgeführt wird. Muss bei diesen noch die notwendige Klasse übergeben werden. Als letztes wird der Player neu geladen.

```
$(".video-js").load();
$(".video-js-box").removeClass("vjs-playing");
$(".video-js-box").addClass("vjs-paused");
```

Am Schluss müssen die Funktionen, in das jeweilige Modul eingebunden werden. Hier ist die Einbindung der bildergalerie.js in das Modul Bildergalerie 2011 zu sehen.

```
<script

    type="text/javascript"
    src="files/js/bildergalerie.js">

</script>
```

Beitragswechsel

Damit ein Blättern zwischen den Eintragsseiten möglich ist. Wird die Funktion `changeEntries()` benötigt, diese ist in der programmplan.js zu finden. Diese Funktion beinhaltet die Manipulation der Eintragsseiten durch die Navigationspfeile.

Damit nicht alle Eintragsseiten angezeigt werden, wird mittels `hide()`, die nachfolgenden Geschwisterknoten, des ersten Elementknotens ausgeblendet.

```
$(".entryPage:gt(0)").hide();
```

Da keine vorige Eintragsseite vorhanden ist, muss dem Element, was zuständig ist für den Wechsel zu der vorigen Seite, die Klasse noPrevPage zugewiesen werden.

```
$(".previousPage").addClass("noPrevPage");
```

Nun muss den beiden Navigationspfeilen, das `click` Event hinzugefügt werden.

```
prevPage.click(function () {

…Ereignisse…

 });
```

Damit die Pfeile wieder optisch sichtbar sind, für die Navigation, muss die Klasse noNextPage beim Zurückblättern und noPrevPage bei den Vorblättern, entfernt werden.

```
nextPage.removeClass("noNextPage");
prevPage.removeClass("noPrevPage");
```

Danach wird überprüft, ob beim Zurückblättern, der nächste Eintragsseitenindex, klein oder gleich des letzten Eintragsindex ist. Falls dies der Fall ist, werden folgende Ereignisse geschaltet. Andernfalls wird nichts ausgeführt.

```
if ( $('.entryPage:visible').next().index() <=
$('.entryPage:last').index())

    { … Ereignisse… }

else { }
```

Bei den Vorblättern wird überprüft, ob der vorige Eintragsindex kleiner als der erste Eintragsindex ist. Ist dies nicht der Fall, werden die Ereignisse ausgeführt, ansonsten geschieht nichts.

```
if ( $('.entryPage:visible').prev().index() <
$('.entryPage:first').index()) { }

else{ …Ereignisse… }
```

Folgende Ereignisse werden geschaltet, nachdem in eine Richtung geblättert wurde. Zuerst wird die derzeitige sichtbare Eintragsseite mittels `fadeout()` ausgeblendet. Danach wird die nächste oder vorige Eintragsseite eingeblendet.

```
$(".entryPage:visible")
      .fadeOut("fast", function (){

          $(this)
               .next(".entryPage")
               .fadeIn("slow");

      });
```

In diesem Beispiel wird nach vorne geblättert, beim zurückgeblättert wird next() durch prev() ersetzt. Beim Klicken wird gleichzeitig überprüft, ob eine Folge-Seite vorhanden ist oder nicht. Ist dies nicht der Fall, wird dem Navigationselement eine Klasse zugeordnet.

```
if ($('.entryPage:visible').next().index() ==
$('.entryPage:last').index()){
     nextPage.addClass("noNextPage");
}
```

Sobald keine vorige Seite vorhanden ist:

```
if($('.entryPage:visible').prev().index()==
$('.entryPage:first').index()) {

     $(".previousPage").addClass("noPrevPage");}
```

3.2.5.3 PHP – Anpassungen

Headertitelausgabe

Damit der rechte Inhaltsbereich in der Headline, immer den richtigen Seitenamen anzeigt, wird mit PHP die aktuelle ID ausgelesen und dem entsprechenden Namen zugeordnet.

```php
<?php

function idAuslesen() {
  $article_id = 1;
  $farbe = array(1 => "red", 2 => "blue", 3 => "green", 4 => "grey" );
  $page_id = $_GET['article_id'];

  if(!$page_id) {
    return $farbe[1];
  }
  else {
    return $farbe[$page_id];
   }
}

function headerTitel() {
    $color =  idAuslesen();
    switch ($color) {
            case "red":
                echo "Beiträge 2011";
                break;
            case "blue":
                echo "Beispiele";
                break;
            case "green":
                echo "Videos";
                break;
                case "grey":
            echo "Sponsoren";
                break;
      }
}
?>
```

Abb. 3.2-11 Quellcode Headertitelausgabe

Manifest aktualisieren

Damit die Ressourcen URLs nicht immer von Hand, in das Manifest eingegeben werden. Wurde ein Script erstellt, was aktuelle Ressourcenpfade ausliest, und dann an einer festgelegten Stelle im Manifest einbindet. Folgende Ressourcen werden für das Manifest benötigt, alle Grafiken aus der Bildergalerie sowie die Grafiken von der Sponsorenseite. Da der Zugriff auf die beiden Ressourcenquellen identisch ist, wird nur ein Beispiel erläutert.

Um auf die Daten zugreifen zu können, muss eine Datenbankverbindung hergestellt werden und ein Selektieren der Daten aus der Tabelle erfolgen.

```
$db_table = "bildergalerie";

$sql = new rex_sql;

$sql->debugsql = 0;

$sql->setQuery("SELECT * FROM $db_table");
```

Nachdem die Daten ausgelesen wurden, muss eine Ausgabe jeder Zeile erfolgen. Die Ausgabe wird dann der Variable $inhalt zugewiesen, die im Vorhinein definiert wurde.

```
$inhalt = " ";
```

Mit einer for-Schleife, wird die Zeilenausgabe realisiert, die dafür sorgt, dass jeder Zeileninhalte, zur Variable `$inhalt` hinzugefügt wird bis alle Zeilen ausgelesen wurden.

```
for($z=0;$z<$sql->getRows();$z++)

{

  $inhalt .= sql->getValue("mediendatei");

  $sql->counter++;

}
```

Jetzt stehen alle URLs der Ressourcen zur Verfügung und müssen in die Manifest Datei eingebunden werden. Um dies zu erreichen, muss das aktuelle Manifest ausgelesen werden, was mit `fopen()` Funktion von php erledigt wird. Da eine Datei ausgelesen wird, muss `fopen()`, der Modi rb zugeordnet werden. Wobei das r für read und das b für nicht Textdateien steht. In diesen Fall, wird die cache-manifest.manifest Datei ausgelesen und dessen Inhalt in der `$manifestContent` Variable zugeordnet. Und danach wird der Zugriff auf die Ressource beendet.

```
$dateiname = './cache-manifest.manifest';
```

```
$handle = fopen($dateiname, 'rb');

$manifestContent

 = strip_tags(stream_get_contents($handle));

fclose($handle);
```

Damit keine Umbrüche als HTML-Code dargestellt werden, wird `strip_tags()` verwendet, diese entfernt alle HTML Elemente. Da jetzt, der Inhalt des Manifests und die verwendeten Ressourcen-URLs bekannt sind. Werden die veralteten URL-Bereiche in dem Manifest, durch die aktuellen URLs ersetzt. Hierfür wird die Funktion `preg_replace()` verwendet, die eine Variable mit einem bestimmten Suchmuster durchsucht. Und sobald das Suchmuster gefunden wurde, wird der vorhandene Abschnitt, mit den neuen Inhalten ersetzt. Sieht folgendermaßen aus:

```
$newManifestContent = "IMGSTART \r\nCache:"

                        . $inhalt .

                        "\r\n# IMGEND";

$suchmuster = "/IMGSTART.*IMGEND/s";

$text = preg_replace(

          $suchmuster,

          $newManifestContent,

          $manifestContent );
```

Nachdem ein neuer String, mit aktuellen Inhalten erzeugt wurde, muss dieser nur noch in das Manifest geschrieben werden. Dies erfolgt wie beim Auslesen des Manifests mittels `fopen()`, nur das diesmal der Modi auf w gestellt wird und somit Inhalte in die Datei geschrieben werden können. Mit `fwrite()` wird dann die Datei mit neuen Inhalten überschrieben. Am Ende wird die Verbindung geschlossen.

```
$writehandle = fopen($dateiname, 'w');
```

```
fwrite($writehandle, $text);

fclose($writehandle);
```

Damit immer ein aktueller Stand des Manifests vorhanden ist, wird das Script in das Default Template eingebunden.

3.2.6 Benutzergruppenverwaltung

Anzahl der Benutzer beschränkt sich auf zwei Gruppen, einmal die Administratoren, denen es möglich ist, Module, Template und Inhalte zu ändern. Die andere Gruppe sind die Redakteure, denen nur der Zugang zu inhaltlichen Änderungen gegeben ist. Festgelegt werden diese im Backend unter den Menüpunkt Benutzer. Dort sind auch alle vorhandenen Zugänge aufgelistet. Hier können dann neue Benutzer angelegt, vorhandene editiert oder entfernt werden.

Folgende Rechte werden den Redakteuren zu gesprochen:

- Zugriff auf die Struktur
- Zugriff auf die Tabellen der XForm

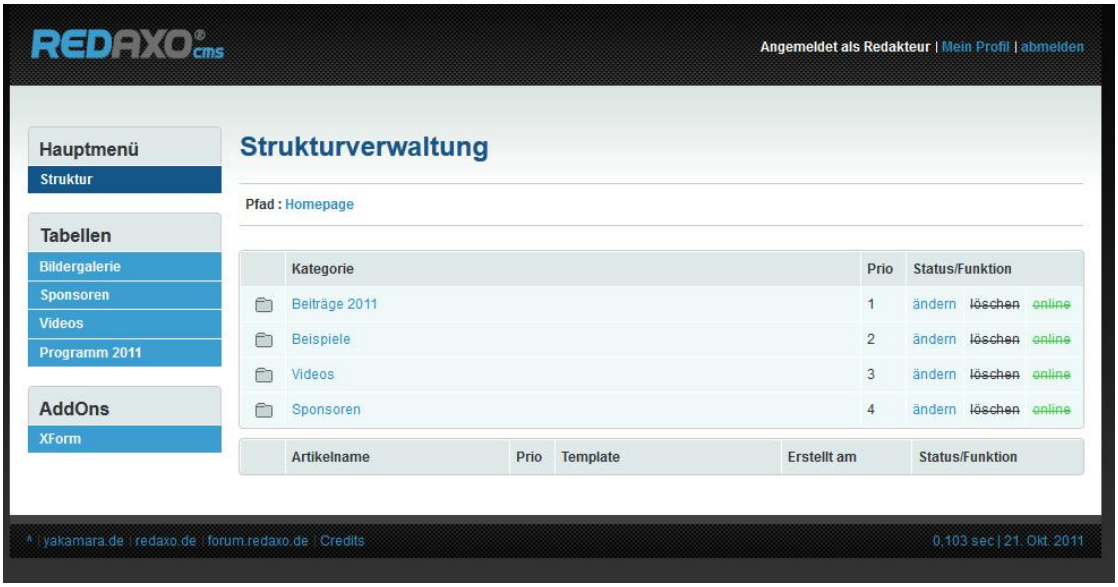


Abb. 3.2-12 Backendansicht für Redakteure

Für Administratoren werden keine Beschränkungen aufgelegt, sie erhalten somit den vollen Zugriff im Backend.

3.2.7 Inhaltseinbindung

In diesem Kapitel wird beschrieben, wie es möglich ist Inhalte über das Backend zu erstellen, zu ändern oder zu löschen.

Es gibt zwei Inhaltsbereiche, in den Inhalte hinzugefügt, bearbeitet oder gelöscht werden können. Einmal den linken Inhaltsbereich, der über die Tippbox und Infobox verfügt, dieser kann im Backend über den Menüpunkt Struktur geändert werden. In der gewünschten Kategorie, und nach Auswahl des Artikels, wird eine Übersicht über vorhandene Einträge und Spalten sichtbar(Abb. 3.2-13).

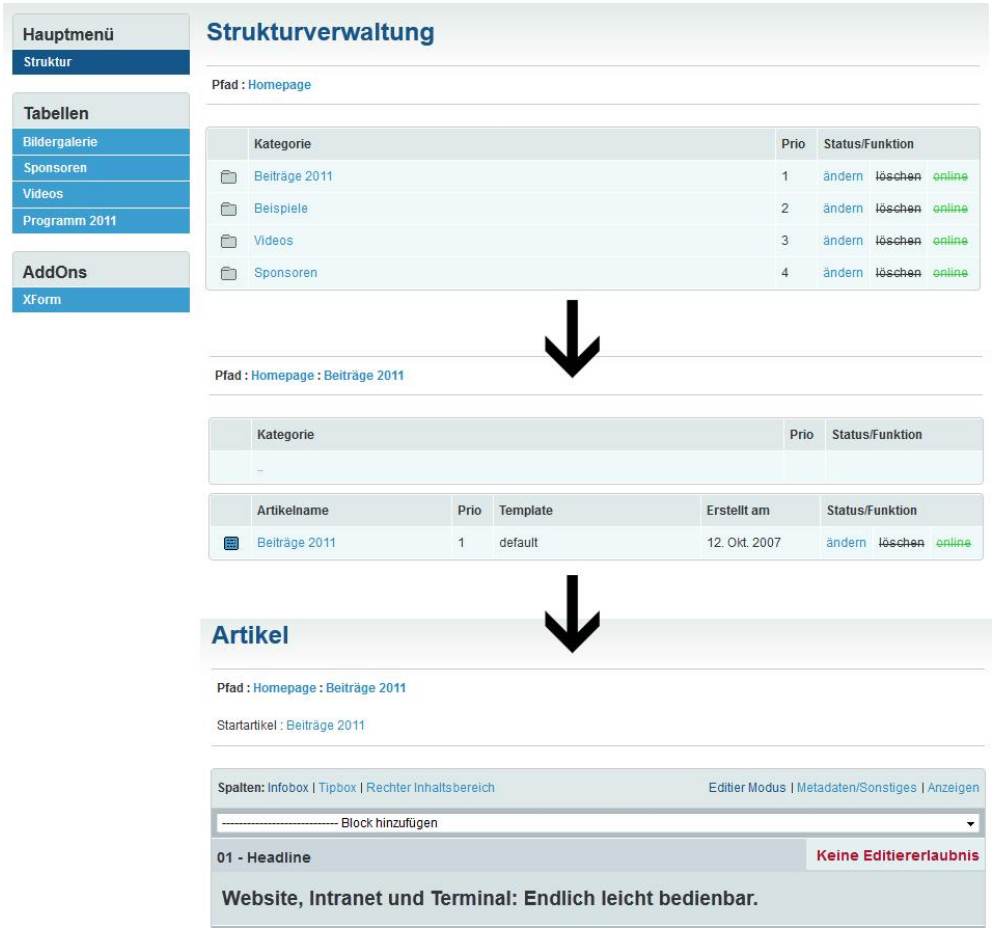


Abb. 3.2-13 Inhaltseinbindung Struktur

Die Webanwendung verfügt über folgende Spalte, die Tippbox, Infobox und der Rechter Inhaltsbereich. Zum Einfügen der Inhalte, in den einzelnen Spalten werden Module verwendet. Bei der Tippbox und Infobox, stehen folgende 2 Module zur Verfügung, für die Überschrift das Modul Headline und für Texte das Modul Text[textile].

Im rechten Inhaltsbereich, werden die Modulausgaben der XForm eingebunden. Wie zum Beispiel das Modul Programm 2011, welches nur über eine Ausgabe des Programmplans verfügt. Da diese Module über keine Eingabe verfügen, ist die Änderung der einzelnen Inhalte über die Tabellen der XForm möglich. Um diese Inhalte zu ändern, muss in der Hauptüberschrift Tabellen im Backend, die zu veränderte Tabelle ausgewählt werden. Danach öffnet sich eine Vorschauseite, der bis jetzt erstellten Inhalte.

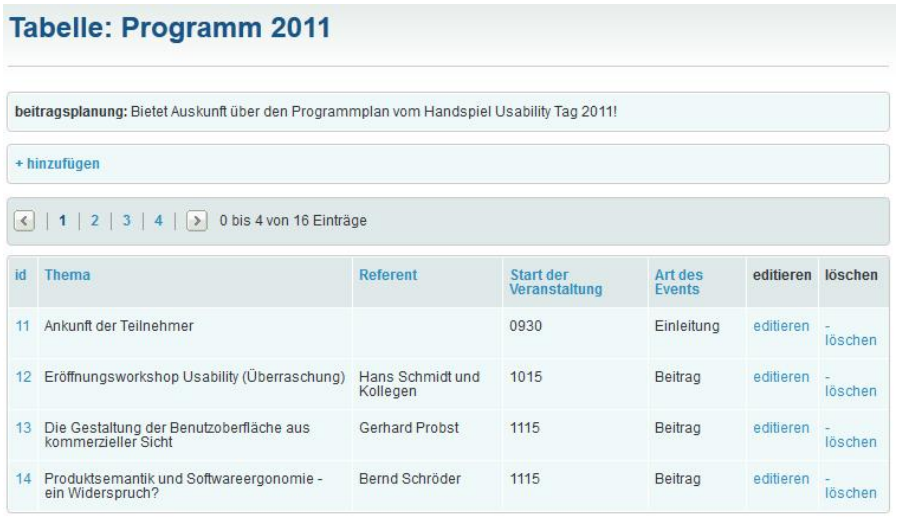


Abb. 3.2-14 Vorschau der Beiträge des Programmplans

Hie könne Inhalte gelöscht, bearbeitet oder neue Inhalte hinzugefügt werden. Beim Bearbeiten und hinzufügen, können Inhalte über ein Formularfeld geändert werden.

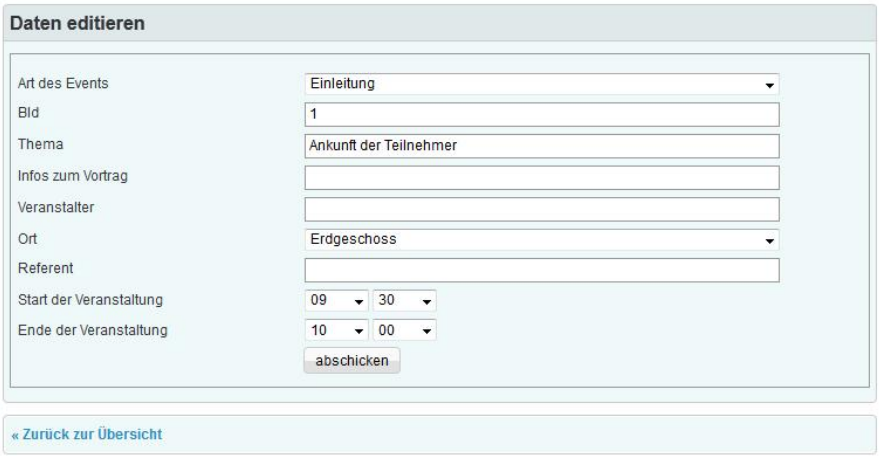


Abb. 3.2-15 Bearbeitung Eintrag

3.3 Maßnahmen zur Umsetzung der Offlinefähigkeit

Jeder kennt das, die Internetverbindung ist weg und die aktuelle geöffnete Webseite funktioniert nicht mehr. Das ist auch der Grund warum bestimmte Desktopprogramme (z.B. Grafikeditor oder Programme zur Textverarbeitung.), noch nicht den Weg ins WWW[1] gefunden haben. Dies könnte jetzt mit HTML5 und diversen erweiternden Technologien sich ändern. Denn mit HTML5, ist es möglich, ihre Webanwendung für den Offline-Betrieb tauglich zu machen. In den nachfolgenden Kapiteln wird aufgezeigt, wie dies möglich ist. (6)

3.3.1 Browserunterstützung

Bei der Browserunterstützung zeichnet sich ein einheitliches Bild ab, so unterstützen die bekannten Browser Firefox, Opera, Chrome und Safari den Web Storage und Offline-Anwendungen. Ausnahme macht der Internet Explorer dieser unterstützt nur den Web Storage ab Version 8.0 und Offline-Anwendungen gar nicht. Sogar die aktuelle Version 9.0 unterstützt keine Offline-Anwendungen.

	Firefox	Opera	Chrome	Safari	IE
Web Storage	3.0	10.50	3.0	4.0	8.0
Offline-Apps	3.5	10.6	4.0	4.0	

Tabelle 1 Browserunterstützung Offline-Apps und Web Storage

3.3.2 Der Application Cache und das Cache Manifest

„Ein Cache Manifest ist eine Datei, in der Adressen zu Ressourcen (Bildern, Scripte, HTML) festgehalten werden, die der Browser für diese Webseite vorrätig halten muss. Was in einem Manifest steht, wird vom Browser geladen und gespeichert – in etwa wie im normalen Browsercache, aber präzise durch die Webseite kontrollierbar, versioniert und automatisch mit dem Netz synchronisiert. Das Ganze ist weniger volatil als der normale Cache und eine bestimmte Ressource muss nicht

[1] WWW – World Wide Web

erst durch den Besucher aufgerufen werden, bevor sie gespeichert wird. Jedes Manifest definiert damit einen eigenen sogenannten Application Cache für die jeweilige Webanwendung.“ (6)

3.3.2.1 Cache Manifest anlegen

Zuerst muss das Cache Manifest angelegt werden, dafür wird eine normale Textdatei benötigt. Hier werden dann die URLs, der zu speichernden Ressourcen aufgelistet. Die Datei muss UTF-8 kodiert sein und ihre MIME-Type muss text/cache-manifest lauten. Das Manifest muss folgenden Aufbau haben:

```
CACHE MANIFEST

# Die erste Zeile ist Pflicht

  # Für weiter Kommentare das Rautenzeichen verwenden

  # Leere Zeilen vor den Kommentar werden ignoriert

  # Gespeicherte Dateien für Offlinebetrieb

  # Pro Zeile eine URL oder Kommentar

# SEITEN

CACHE:

./index.php?article_id=1

./index.php?article_id=2

./index.php?article_id=4


# JAVASCRIPT

CACHE:

./files/js/jquery-1.6.1.min.js

…
```

Alle eingetragenen Pfade müssen relativ zum Manifest angegeben werden. Die Einbindung des Manifests erfolgt über das html-Element, notwendig ist dafür ein relativer Pfad.

```
<html manifest="./cache-manifest.manifest">
```

Um die Funktionsfähigkeit zu testen, könnte die Netzverbindung getrennt werden und danach muss die Seite erneut aufgerufen werden. Nun ist aber nicht erkennbar, ob Fehler in der Manifest Datei vorhanden sind oder ob Dateien im Application Cache abgelegt wurden. Was benötigt wird, ist eine umfassendes Logging aller Cache Aktivitäten, was über die passenden Events machbar ist.

3.3.2.2 Manifest-Events

Um zu überwachen, wie gut unser Manifest funktioniert, biete HTML5 eine API für den Application Cache. In diesen befinden sich folgende Events:

Event	Ereignis	Folge-Events
`Checking`	Browser versucht das Manifest zum ersten Mal zu laden oder prüft das Manifest auf Updates	`Noupdate`, `downlaoding`, `absolete`, `error`
`Noupdate`	Das Manifest hat sich nicht verändert,	`Keine`
`Downloading`	Das Manifest wird heruntergeladen oder seine Ressourcen werden erstmals heruntergeladen	`Progress`. `Error`, `cached`, `updateready`
`Progress`	Ressourcen werden heruntergeladen	`Progress`, `error`, `cached`, `updateready`

Cached	Ressourcen werden heruntergeladen	Keine
Updateready	Ressourcen wurden aktualisiert	Keine
Absolete	Manifest Anfrage lieferte Fehler 404 oder 410, Application Cache wird gelöscht	Keine
Error	Das Manifest oder die Webseite konnte nicht korrekt geladen werden, Fehler beim Download der Ressourcen oder das Manifest hat sich während des Downloades verändert	Keine, der Browser wird, hat sich nur das ManifestInhalt geändert, erneut einen Download versuchen

Tabelle 2 Manifest-Events(6)

Jetzt müssen nur noch die Events, in die JavaScript Konsole eingeloggt werden, dies erfolgt über die JavaScript-Datei mf.js. Hier werden die Events des Application Cache durch einen Eventlistener abgefangen. Und per log der Zustand des aktuellen Events in der Konsole wieder gegeben. Das nachfolgende Beispiel zeigt, wie das checking-Event eingeloggt wird und je nach Ergebniss, Folge Events schaltet.

```
if(typeof applicationCache !== 'undefined'){

applicationCache.addEventListener('checking',
function(){

   console.log('Suche Manifest...');

}, false);
```

Würde zum Beispiel das Manifest einen Fehler enthalten, so würde als Nächstes ein error-Event geschaltet werden. Dies wird abgefangen und in der Konsole eine Fehlermeldung ausgegeben.

```
applicationCache.addEventListener('error', function(){
   console.log('Fataler Fehler bei anlegen das Caches');
}, false);
```

Wäre die Suche erfolgreich, wird eine Ressource nach der anderen in den Cache gespeichert. Sobald aber ein Fehler auftritt, wird die Speicherung abgebrochen. Siehe nachfolgende Abb. 3.3-1.

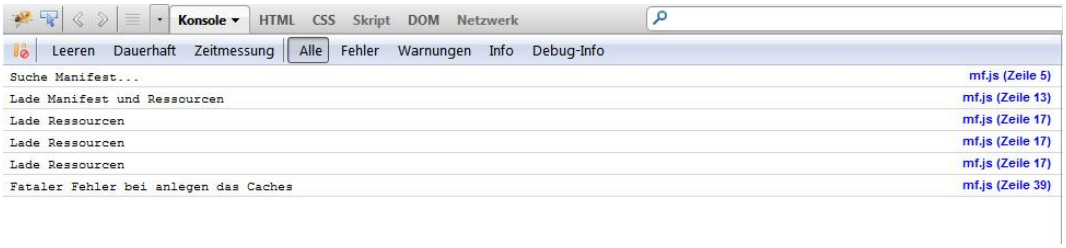


Abb. 3.3-1 Konsolenausgabe der Events

Sobald keine Fehlermeldung erscheint und der Application Cache bereit ist. Nun kann durch Trennung der Internetverbindung, die Anwendung auf ihre Offlinefähigkeit getestet werden. Indem man den Webserver ausschaltet, den Netzwerkstecker zieht oder durch den Firefox unter Datei, in den „Offline arbeiten“ Modus wechselt. Wenn alle Ressourcen URLs korrekt in das Manifest eingetragen sind und somit alle Dateien in den Offline Cache geladen wurden, müsste die Webanwendung, im vollen Umfang zu sehen sein.

3.3.2.3 Fallbacks und Whitelists

Um zu verhindern, dass Daten in den Offline Cache gespeichert werden, wird eine Online Whitelist verwendet. Notwendig wird die Whitelist wenn zum Beispiel nur bestimmte Inhalte für den Online Betrieb, zur Verfügung gestellt werden sollen. Zum Beispiel für einen Newsticker anwendbar, da der nur neueste Informationen darstellen soll und ältere Information unpassend wären.

So wird allen Ressourcen die nicht gespeichert werden sollen, NETWORK: vorangestellt. Für den Newsticker als Beispiel, sieht der Eintrag im Manifest, dann so aus:

```
NETWORK:

newsticker.html
```

Nun wird beim Aufruf der Seite newsticker.html, eine Fehlermeldung ausgegeben. In unseren Fall wär das eine „Seite nicht gefunden**“** Meldung. Um die Fehlermeldung zu vermeiden, gibt es Fallback-Inhalte, die aufgerufen werden, sobald eine Ressource nicht vorhanden ist. Bei der Handspiel Webanwendung wird dann zum Beispiel, die Startseite stattdessen angezeigt. Wichtig, es ist darauf zu achten, dass der Ressource ein Slash vorausgesetzt wird.

```
# Fallback

FALLBACK:

/ ./index.php?article_id=1
```

3.3.2.4 Überprüfung Offline Cache und dessen Leerung

Um genauere Informationen über die aktuellen gespeicherten Ressourcen im Browser Cache zu erhalten. Bietet der Firefox eine Möglichkeit, Informationen über den Cache, direkt im Browser darzustellen. Um diese Informationen zu erhalten, muss die Adresse about:cache aufgerufen werden. Unter Offline-Cache-Device werden alle Ressourcen angezeigt, die für den Offlinebetrieb gespeichert wurden. Bei Auswahl einer Ressource, können zusätzliche Informationen dargestellt werden.

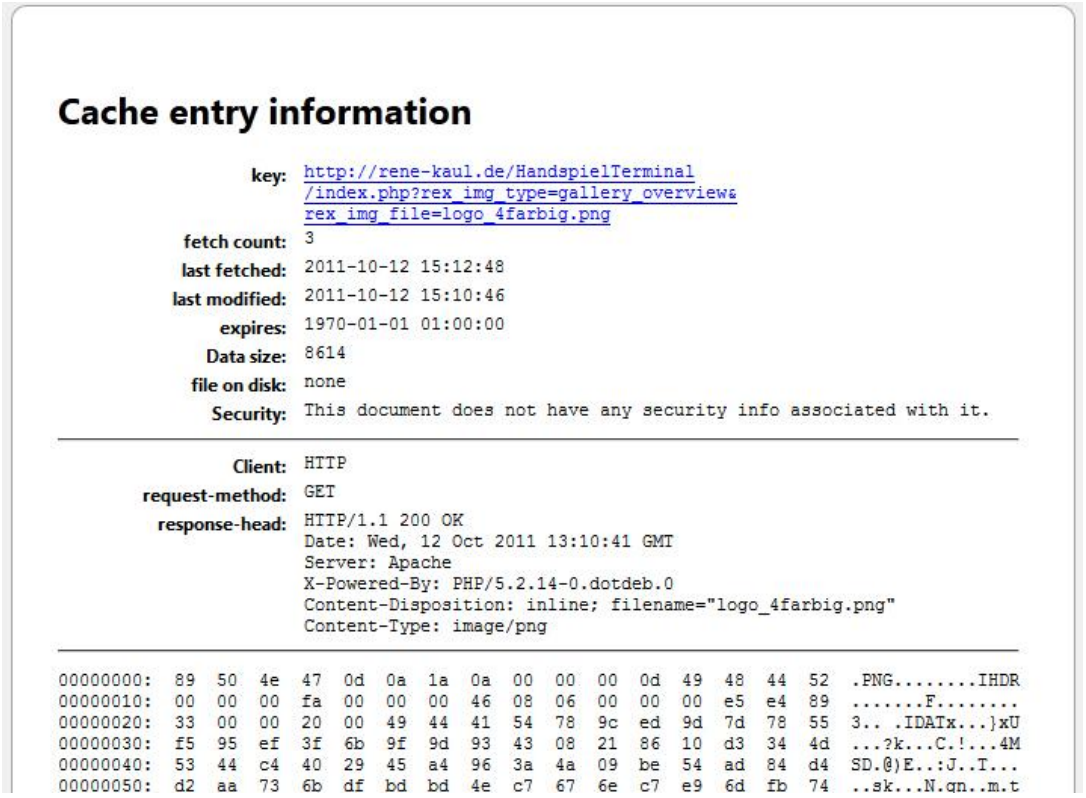


Abb. 3.3-2 Cache Entry Information

Die Leerung des Offline Cache erfolgt beim Firefox über die Einstellungen, diese findet man in der Navigation unter Extras. Unter „Erweitert“ im Reiter Netzwerk, kann man den Cache jetzt leeren und die einzelnen Webseiten aus dem Cache löschen, siehe Abb. 3.3-3.

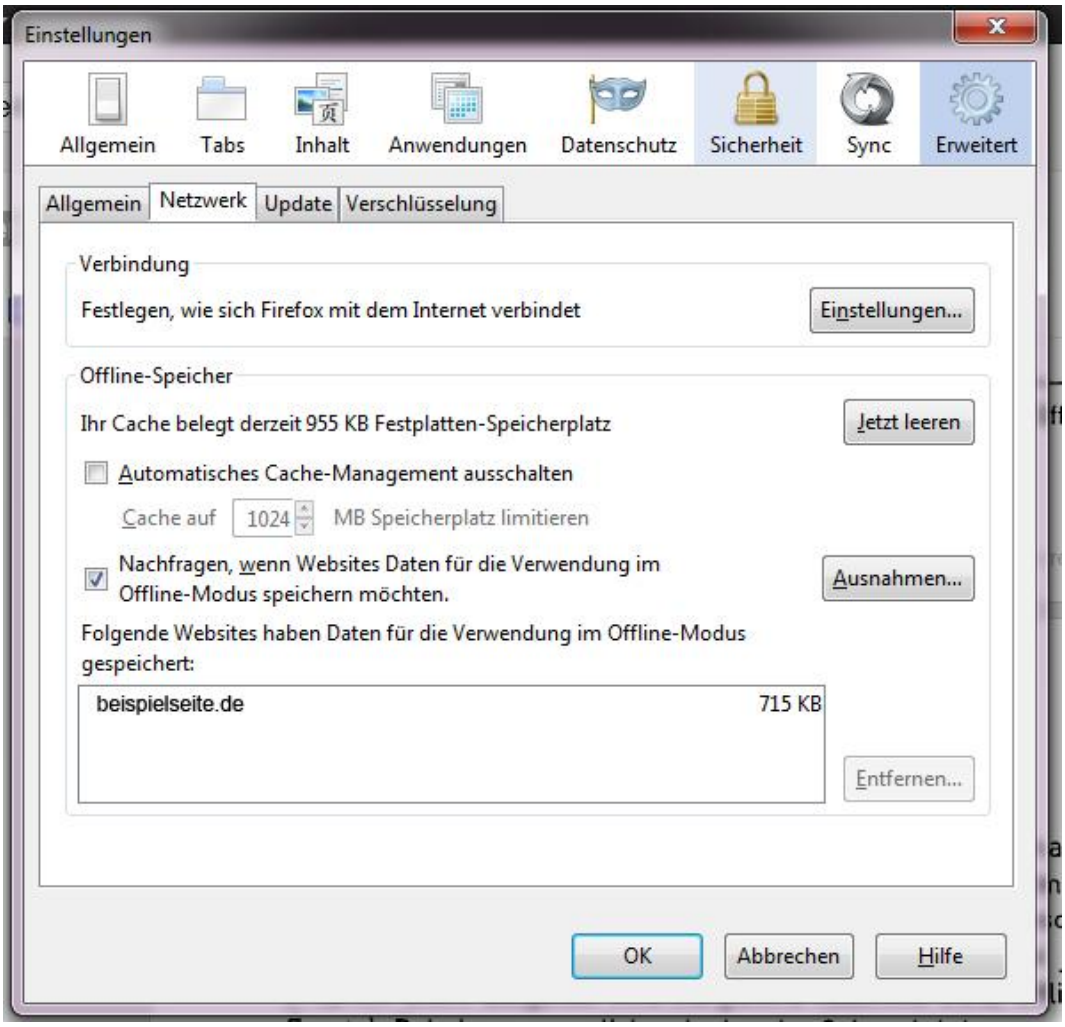


Abb. 3.3-3 Einstellungen Firefox

Eine weitere Möglichkeit besteht darin, die lokalen Dateien zu löschen. So finden sich die Dateien für den Application Cache unter Windows Vista / 7 beim Firefox unter folgenden Pfad:

```
C:\Users\<benutzername>\AppData\Local\Mozilla\Firefox\Profiles\<profile|\OfflineCache
```

Und beim Windows XP unter:

```
C:\Dokumente und Einstellungen\<benutzername>\Lokale einstellungen\Anwendungsdaten\Mozilla\Firefox\Profiles\<profile|\OfflineCache
```

3.3.3 DOM Storage und Online-/Offline-Events

DOM Storage bietet die Möglichkeit, Name-Wert-Paare im Browser lokal zwischenzuspeichern. So können zum Beispiel, Inhalte die durch eine Nutzer erzeugt wurden, bei nicht vorhandener Verbindung zum Server, zwischen gespeichert werden. Bei Wiedereherstellung der Verbindung, kann der vorhandene Datenbestand, mit dem Server synchronisiert werden. Informationen über den Status der Verbindung, sind erhältlich über die Online-/Offline Events die immer feuern, sobald die Verbindung abbricht oder wieder hergestellt wird.

In unserer Anwendung kommt der DOM Storage nicht zum Einsatz, da keine notwendigen Daten vom Nutzer erzeugt werden. Aber könnte ein sinnvoller Erweiterungspunkt für das Backend sein. Siehe dafür Kapitel 6.

4 Anwendungstest

Beim Anwendungstest wurden alle Richtlinien und Vorgaben sowie Funktionsweisen der Anwendung, durch die Mitarbeiter der Handspiel GmbH getestet und überprüft.

Ergebnisse des Tests waren:

Die problemlose Ausführung der Webanwendung, im Firefox und auf den Betriebssystem Windows 7 wurde bestätigt. Bei der Darstellung im Frontend ist eindeutig zu erkennen, dass es sich um ein Produkt der Handspiel GmbH handelt. Die Textdarstellung und die eingefügten Bilder sind gut erkenntlich und lesbar. Das Navigieren zwischen den einzelnen Inhaltsseiten funktioniert fehlerfrei. Durch den farblichen Rahmen und der Überschrift, im rechten Inhaltsbereich, ist eine eindeutige Identifizierung, der aktuellen Inhaltsseite gegeben. Die aktuelle Zeit wird fehlerfrei auf jeder Inhaltsseite angezeigt und die Ausführung der Webseite im Offlinebetrieb ist gegeben.

Alle Inhalte, der einzelnen Inhaltsseiten, sind in einer ordentlichen Struktur dargestellt. Zusätzlich zu den Inhalten, wurde jede einzelne Seite auf ihre Funktionalitäten überprüft. Auf der Beitragsseite ist das Navigieren zwischen den einzelnen Beiträgen gegeben und funktioniert fehlerfrei. Des Weiteren ist bei den Navigationspfeilen erkenntlich, wenn keine vorige oder folge Einträge vorhanden sind. Auf der Beispielseite funktioniert die Bildergalerie einwandfrei und nach der Auswahl eines Miniaturbildes, wird das Hauptbild und dessen Beschreibung ordnungsgemäß gewechselt. Der eingerichtete Videoplayer auf der Videoseite funktioniert einwandfrei und ermöglicht eine Steuerung der Videos. Die hinzugefügte Navigation stellt alle aufgelisteten Medien, bei einer Auswahl, ordnungsgemäß zur Verfügung.

Über den Redakteur-Zugang war es möglich, Inhalte zu bearbeiten, zu löschen oder neue Inhalte zu erzeugen und es ist erkenntlich, in welchen Kategorien sich die jeweiligen Inhalte befinden.

5 Zusammenfassung

Ziel der Bachelorarbeit war es, für die Handspiel GmbH eine offlinefähige Webanwendung zu konzipieren und zu entwickeln. Der Einsatzbereich sollte sich dabei auf Terminals oder Touchscreen-Monitore beschränken. Als inhaltliche Orientierung diente der HANDSPIEL-Usability-Tag 2011.

Um eine Grundlage für die Entwicklung zu schaffen, befasst sich Kapitel 2, mit der Konzipierung der Webanwendung. Es wurden Richtlinien und Vorgaben festgelegt, ein geeignetes Content Management-System gefunden und verwendete Technologien geprüft. Bei der Prüfung ergab sich, das HTML5 als geeignete Technologie, für die Umsetzung der Offlinefähigkeit, verwendbar ist.

Im praktischen Teil der Arbeit wurden Themen wie, die Gestaltung des Frontends und die gesamte Implementierung der Webanwendung, abgehandelt. Die Implementierung beinhaltete, die Einrichtung eines CMS und die Erstellung notwendiger Module, Templates, Funktionen, CSS-Dateien. Dabei kamen folgende Technologien zum Einsatz, HTML, CSS, jQuery, JavaScript, MySQL und PHP. Am Ende der Implementierung wurden notwendige Maßnahmen ergriffen, um eine Offlinefähigkeit der Webanwendung, zu bewerkstelligen. Als Abschluss wurde die gesamte Anwendung, durch die Handspiel GmbH, auf ihre Funktionalitäten geprüft.

Insgesamt ist eine Webanwendung entstanden, die alle notwendigen Anforderungen und Funktionalitäten beinhaltet, die im Vorhinein festgelegt wurden. Es wurde aufgezeigt, dass es möglich ist, ohne vorhandene Internetanbindung, die Webanwendung funktionsfähig zu halten. Unter der Bedingung, dass die Daten vorher in den Offline-Cache gespeichert wurden. Somit ist die Anwendung betriebsbereit und kann für den HANDSPIEL-Usability-Tag 2011 verwendet werden.

6 Weiterführung der Arbeit

In folgenden Bereichen könnte die Arbeit weitergeführt werden.

Bisher wird bei einem Seitenwechsel, der gesamte Inhalt neu geladen. Um Verbesserung in der Performance zu erhalten, könnte die Anwendung so ausgebaut werden, dass nur Inhaltsbereiche neu geladen werden, die auch tatsächlich geändert werden. Die Umsetzung könnte zum Beispiel mit der Verwendung von Ajax erfolgen.

Da die Webanwendung primär für FullHD-Touchscreen konzipiert wurde und die gesamte Anwendung eine feste Größe hat, wäre ein Erweiterungspunkt, die auflösungsunabhängige Darstellung. So könnten verschiedene Darstellungsversionen entwickelt werden, die je nach Auflösung angezeigt werden.

Primär wurde in der Arbeit, eine Webanwendung erstellt, die am HANDSPIEL-Usability-Tag 2011 zum Einsatz kommt. So wurden auch die Module speziell hierfür erstellt. Für weitere Präsentationsbereiche, wäre es sinnvoll, eine Modulbibliothek anzufertigen, um das variieren der Inhalte zu ermöglichen. Sinnvoll wäre auch eine Erweiterung der Webanwendung, mit verschiedenen Sprachversionen, um eine kulturelle unabhängige Bedienbarkeit und Darstellung zu ermöglichen.

Bisher ist es der Webanwendung möglich, auch ohne Internetverbindung ihre Inhalte darzustellen. Da aber das Backend, nicht offlinefähig ist und somit ein Ändern der dargestellten Informationen durch den Redakteur nicht gegeben ist. Wäre ein Erweiterungspunkt, die Umsetzung eines offlinefähigen Backends.

Literaturverzeichnis

(1) Fischer, Mario: Website Boosting 2.0. – 2.Aufl. - Heidelberg, München, Landsberg, Frechen, Hamburg : Hüthig Jehle Rehm GmbH, 2009

(2) Google: Über Gears. URL: http://gears.google.com/support/bin/answer.py?hl=de&answer=79873, verfügbar am 19.10.2011

(3) Grammiweb: Was ist JavaScript?. URL: http://www.grammiweb.de/informativ/grundlagen/wasistjava.shtml, verfügbar am 22.10.2011

(4) Gutkovski, A. ; Derchlser, D. ; & Schubert, A. : Webdesign mittels HTML5 und CSS(3). URL: http://winfwiki.wi-fom.de/index.php/Webdesign_mittels_HTML5_und_CSS%283%29#Einleitung, verfügbar am 22. 10. 2011

(5) Kristinus, J. : REDAXO - URL: http://www.redaxo.org/, verfügbar am 11.10.2011

(6) Kröner, Peter: HTML5 Webseiten innovativ und zukunftsicher. – 1.Aufl. - München : Open Source Press, 2010

(7) Meyer, Robert: Praxiswissen TYPO 3. Köln: O'Reilly. - 4.Aufl. - Köln : O'Reilly, 2010

(8) Münz, Stefan. ; Nefzger, Wolfgang: HTML & Web-Publishing Handbuch. - 1.Aufl. . Poing: Franzis´ Verlag GmbH, 2002

(9) PHP: PHP-Handbuch. URL: http://www.php.net/manual/de/preface.php, verfügbar am 18.10.2011

(10) Vollendorf, Maximilian ; Bangers, Frank: jQuery Das Praxisbuch . - 1.Aufl. . Bonn: Galileo Press, 2010

Anhang

Alle Anhänge befinden sich auf der beiliegenden CD-ROM.

Anhang A

Quelltexte

- Die Quelltexte der CSS-Dateien
- Der Quelltext der Templates
- Der Quelltext der Module
- Der Quelltext von JavaScript-Dateien
- Der Quelltext des Manifest

Anhang B

Digitale Umsetzung des Layouts als PSD-Datei

Selbständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Leipzig, den 1.11.2011

René Kaul